

# **Numerische Sensitivitätsanalyse unstetiger multidisziplinärer Modelle mit Anwendungen in der gradientenbasierten Optimierung**

Dissertation

zur Erlangung des akademischen Grades  
doctor rerum naturalium (Dr. rer. nat.)

vorgelegt der

Naturwissenschaftlichen Fakultät III  
für Agrar- und Geowissenschaften, Mathematik und Informatik  
der Martin-Luther-Universität Halle-Wittenberg

von Andreas Pfeiffer

Gutachter:

1. Prof. Dr. M. Arnold, Martin-Luther-Universität Halle-Wittenberg
2. Prof. Dr. C. Führer, Lund University (Schweden)

Eingereicht am 17.03.2008.

Verteidigt am 26.06.2008.

Veröffentlicht in PFEIFFER, A.: *Numerische Sensitivitätsanalyse unstetiger multidisziplinärer Modelle mit Anwendungen in der gradientenbasierten Optimierung*. Fortschrittsberichte VDI, Reihe 20, Nr. 417, VDI Verlag, Düsseldorf, 2008. ISBN 978-3-18-341720-9.



# Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit am Institut für Robotik und Mechatronik im Deutschen Zentrum für Luft- und Raumfahrt e. V. (DLR). Dem Leiter meiner Abteilung Systemdynamik und Regelung, Herrn Dr. Johann Bals, danke ich für die wissenschaftliche Aufgabenstellung, die stetige Unterstützung und den gewährten zeitlichen Freiraum zur Ausarbeitung der Dissertation.

Die zahlreichen fachlichen Diskussionen mit Herrn Prof. Martin Arnold waren eine wichtige Säule für die Entstehung der Arbeit. Für seine freundliche und geduldige Betreuung, die ausführliche Durchsicht des Entwurfsexemplars sowie die Übernahme des Erstgutachtens danke ich ihm recht herzlich. Das Zweitgutachten verfasste Herr Prof. Claus Führer. Für sein Interesse an der Arbeit und seine inhaltlichen Anregungen bin ich ihm sehr dankbar.

Optimierungsthematiken diskutierte ich gerne mit Herrn Dr. Hans-Dieter Joos, der mir mit seinen sachkundigen Hinweisen dankenswerterweise oft weiterhalf. Herr Prof. Martin Otter war der Ansprechpartner, wenn Modelica-spezifische Fragen auftauchten. Für seine hilfreichen Antworten danke ich ihm. Allen anderen Kolleginnen und Kollegen aus der Abteilung – auch den ehemaligen! – bin ich für fachliche Gespräche und Tipps sowie die gute Arbeitsatmosphäre sehr dankbar.

Oberpfaffenhofen, im Juli 2008

Andreas Pfeiffer



# Inhaltsverzeichnis

<b>Nomenklatur</b>	<b>VII</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation und Stand der Technik . . . . .	1
1.2 Zielstellung und Aufbau der Arbeit . . . . .	5
<b>2 Unstetige multidisziplinäre Modelle</b>	<b>7</b>
2.1 Multidisziplinäre Modellierung mit Modelica . . . . .	7
2.1.1 Die Modellierungssprache Modelica . . . . .	8
2.1.2 Unstetige Modellierungselemente in Modelica . . . . .	10
2.2 Umformung der Modellgleichungen . . . . .	13
2.2.1 Struktur der Gleichungen . . . . .	13
2.2.2 Numerische Zustandsform der Modellgleichungen . . . . .	16
<b>3 Lösungstheorie unstetiger Systeme</b>	<b>20</b>
3.1 Schaltsituationen am Ereignispunkt . . . . .	20
3.1.1 Klassische Lösung . . . . .	20
3.1.2 Filippov-Lösung . . . . .	22
3.1.3 Beispiele unstetiger Systeme . . . . .	26
3.2 Besonderheiten an Ereignispunkten . . . . .	30
3.2.1 Nichtglatte Schaltfunktionen . . . . .	30
3.2.2 Mehrere aktive Schaltfunktionen bei rekursiver Staffelung . . . . .	38
<b>4 Sensitivitätsanalyse unstetiger Systeme</b>	<b>42</b>
4.1 Ableitungen der Lösung nach Parametern . . . . .	42
4.1.1 Sensitivitätsgleichungen glatter Systeme . . . . .	43
4.1.2 Sensitivitätsgleichungen hybrider Systeme . . . . .	45
4.1.3 Adjungierte Gleichungen hybrider Systeme . . . . .	51
4.1.4 Parametersensitivitäten der Filippov-Lösung . . . . .	54
4.2 Gleichzeitig aktive Schaltfunktionen . . . . .	56
4.2.1 Problemanalyse . . . . .	56
4.2.2 Beispiele . . . . .	63
<b>5 Numerische Sensitivitätsanalyse unstetiger Modelica-Modelle</b>	<b>68</b>
5.1 Ansätze zur numerischen Sensitivitätsberechnung . . . . .	68
5.1.1 Externe und interne numerische Differentiation, adjungierte Variablen	69
5.1.2 Integration der Sensitivitätsgleichungen mit DASPK3.1 . . . . .	71
5.2 FISEMO: Filippov-Lösungen und Sensitivitätsanalyse von Modelica-Modellen . . . . .	75

---

5.2.1	Die Softwareumgebung FISEMO . . . . .	76
5.2.2	Numerische Integration von unstetigen Modelica-Modellen . . . . .	79
5.3	Numerische Ergebnisse . . . . .	83
<b>6</b>	<b>Anwendung: Gradientenbasierte Parameteridentifizierung eines Robotermodells</b>	<b>89</b>
6.1	Parameteridentifizierung und numerische Lösungsansätze . . . . .	89
6.1.1	Identifizierung von Modellparametern . . . . .	89
6.1.2	Numerische Lösung des Optimierungsproblems – SQP-Verfahren . . . . .	92
6.2	Numerische Ergebnisse für ein Robotermodell . . . . .	94
6.2.1	Dynamisches Robotermodell in Modelica . . . . .	94
6.2.2	Optimierungs- und Identifizierungsergebnisse . . . . .	96
<b>7</b>	<b>Zusammenfassung und Ausblick</b>	<b>101</b>
<b>A</b>	<b>Programmablaufpläne</b>	<b>103</b>
<b>B</b>	<b>Ergebnisse der Sensitivitätsanalysen</b>	<b>105</b>
<b>C</b>	<b>Ergebnisse der Parameteridentifizierung</b>	<b>116</b>
	<b>Literaturverzeichnis</b>	<b>124</b>

# Nomenklatur

$A$	Funktion, die Zustandsänderungen am Ereignispunkt beschreibt.
ATOL	Absolute Fehlertoleranz.
$C$	Zielfunktion in der Optimierung.
$D^+$	Operator für die Richtungsableitung in den positiven Ast der Schaltfunktion.
$D^-$	Operator für die Richtungsableitung in den negativen Ast der Schaltfunktion.
$e_i$	$i$ -ter Einheitsvektor.
eps	Maschinengenauigkeit.
$f$	Rechte Seite eines Systems gewöhnlicher Differentialgleichungen.
$f_x$	Partielle Ableitung von $f$ nach $x$ .
$g$	Abbildung zur Berechnung der algebraischen Variablen und Ausgabegrößen $y$ .
$I$	Einheitsmatrix.
$\text{id}_x$	Identität bzgl. $x$ .
$L$	Lagrange-Funktion.
$p$	Vektor der Parameter.
$q$	Vektor aller aktiven Schaltfunktionen.
$Q$	Vektor aller Schaltfunktionen.
$\nabla q$	Gradient einer eindimensionalen Schaltfunktion $q$ .
RTOL	Relative Fehlertoleranz.
$s$	Vektor der Schalter der aktiven Schaltfunktionen.
$S$	Vektor der Schalter aller Schaltfunktionen.
$\Sigma$	Schaltfläche.
sgn	Vorzeichen-Funktion mit den Werten $+1, 0, -1$ .
$t$	Unabhängige Variable Zeit.
TOL	Fehlertoleranz, falls $\text{ATOL} = \text{RTOL}$ gilt.
$x$	Zustandsvektor.
$\dot{x}$	Zeitableitung des Zustandsvektors.
$y$	Vektor der algebraischen Variablen und Ausgabegrößen.
□	Ende einer Definition, einer Bemerkung, einer Iteration oder eines Beispiels.
■	Ende eines Beweises oder eines Korollars.





# 1 Einleitung

## 1.1 Motivation und Stand der Technik

Die Entwicklung, Analyse und Bewertung von technischen Systemen und Prozessen stützt sich verstärkt auf simulationsbasierte Untersuchungen. Damit ergeben sich durch den technologischen Fortschritt stets neue Anforderungen bzw. Herausforderungen für zukünftige Simulationssoftware. Eine etablierte Plattform für die Simulation integrierter technischer Systeme bietet die Modellierungssprache *Modelica* [Ott00, CEM<sup>+</sup>02, OEM07, Mod]. Sie zeichnet sich durch die Eigenschaft aus, multidisziplinäre (auch genannt: multiphysikalische) Systeme, wie etwa mechatronische Systeme, einheitlich und übergreifend zu beschreiben. Die Mechatronik vereint mechanische, elektronische und informationstechnische Teilkomponenten eines technischen Systems. Das Antiblockiersystem (ABS) eines Kraftfahrzeugs regelt beispielsweise die mechanischen Bremskräfte auf die Räder elektronisch so, dass die Räder während der Verzögerung des Fahrzeugs nicht stillstehen.

### Unstetige Systeme

Die mathematisch-physikalischen Modellgleichungen führen in Modelica zu einem Anfangswertproblem für Systeme gewöhnlicher Differentialgleichungen. Multiphysikalische Systeme werden häufig durch Differentialgleichungen mit *unstetiger* rechter Seite modelliert. Unstetigkeiten ergeben sich teilweise daraus, dass vereinfachte physikalische Zusammenhänge angenommen werden müssen, um den Detaillierungsgrad des Gesamtmodells ausgewogen zu halten. Beispielsweise macht es in den meisten Anwendungsfällen wenig Sinn, das Schließen eines elektrischen Schalters ausführlich mit Übergangseffekten wie einem Lichtbogen zu simulieren. Die elektrische Spannung an dem Schalter wird deswegen zum Zeitpunkt des Einschaltens unstetig auf Null gesetzt. So kann der Einfluss eines ab- oder zugeschalteten elektrischen Antriebs auf ein mechatronisches System (z. B. ein Elektrofahrzeug) hinreichend genau untersucht werden.

Ähnliche Überlegungen führen in allen physikalischen oder informationstechnischen Teilbereichen zu unstetig modellierten Vorgängen: Öffnen oder Schließen von hydraulischen oder pneumatischen Ventilen, nichtglatte (unstetige) Anregungsfunktionen in Regelungsstrukturen, Coulomb-Reibung oder mechanische Stoßvorgänge. Da sich während eines Stoßes auch Zustände (Geschwindigkeiten) des Systems unstetig ändern, erfordert die Simulation von Stößen weitergehende Änderungen der zu Grunde liegenden mathematischen Struktur. Es ergibt sich eine Folge von Anfangswertproblemen für Systeme gewöhnlicher Differentialgleichungen mit Anfangswerten, die von der Lösung des vorhergehenden Systems abhängen. In vielen Fällen kommen zu den Differentialgleichungen algebraische

Zwangsbedingungen hinzu, so dass insgesamt eine Folge von differential-algebraischen Gleichungssystemen entsteht.

Im Modelica-Modell werden Unstetigkeiten in der rechten Seite durch ein Umschalten zwischen glatten Teilmodellen realisiert. Die Gleichungen der Teilmodelle sind dabei in der Gesamtheit aller Modellgleichungen als Äste von bedingten Verzweigungen enthalten. Das Kraftgesetz für die Coulomb-Reibung [ESF02] sieht demnach mit dem Reibungskoeffizienten  $\mu$  für eine Masse  $m$  wie folgt aus: Wenn für die Geschwindigkeit  $v > 0$  gilt, dann sei die Reibungskraft  $F_R := \mu m$ , ansonsten gelte  $F_R := -\mu m$ . Die Verzweigung in einen der beiden Äste wird nicht direkt über den Wahrheitswert der Relation  $v > 0$  gesteuert, sondern von einem Schalter vorgegeben, der unabhängig vom Wert von  $v$  gesetzt werden kann. Der Ausdruck  $v$  heißt in dem Zusammenhang *Schaltfunktion*. Diese Technik unterscheidet sich davon, das Kraftgesetz direkt mittels  $F_R := \mu m \cdot \text{sgn } v$  zu formulieren.

Das Konzept mit Schaltern und Schaltfunktionen ist seit vielen Jahren bekannt [Boc87, HNW93, Ott94, CK06] und kann als Standard-Ansatz zur numerischen Lösung von gewöhnlichen Differentialgleichungen mit unstetiger rechter Seite bezeichnet werden. Praxis-taugliche Implementierungen von Zeitintegrationsverfahren sind auf diesen Ansatz zugeschnitten und verwenden die Schaltfunktionen zur Lokalisierung der Unstetigkeitsstelle während der Integration. Hierfür ist es nötig, dass die Schalterstellungen aller Schalter im Modell solange konstant gehalten werden, bis der Zeitpunkt der Unstetigkeitsstelle berechnet ist. Die numerische Integration beginnt dann an diesem Zeitpunkt mit neuen Schalterstellungen und der resultierenden rechten Seite von Neuem. So wird eine zuverlässige numerische Lösung des abschnittsweise glatten Differentialgleichungssystems ermöglicht. Zur Modellierungssprache Modelica existieren Programme, die die numerische Lösung der Modellgleichungen mit Schaltfunktionen unterstützen. Der momentan leistungsfähigste Vertreter dieser Programme ist Dymola [Dym].

Ein typisches Problem, das unstetige rechte Seiten mit sich bringen, betrifft die Existenz von Lösungen. In bestimmten Situationen kann an einer Unstetigkeitsstelle keine Schalterstellung gefunden werden, für die sich die Lösung fortsetzen lässt. Hier ist es sinnvoll, das verallgemeinerte Lösungskonzept von Filippov [Fil88] anzuwenden. Als Standard-Beispiel ist die Coulomb-Reibung während der Haftreibungsphase zu nennen. In dieser Lösungsphase ruht die Masse ( $v = 0$ ), auch wenn äußere Kräfte auf sie wirken. Die entsprechenden Lösungen können mit verallgemeinerten Differentialgleichungen, den Differentialinklusionen, formuliert werden. Hierfür werden mengenwertige rechte Seiten betrachtet, die im Spezialfall einer stetigen rechten Seite nur aus einem Element bestehen.

Die Berechnung dieser verallgemeinerten Lösungen wird in Modelica bisher nicht unterstützt. Man behilft sich z. B. für die Coulomb-Reibung dadurch, dass im Modell logische Zustandsautomaten bestimmen, ob die Gleit- oder Haftreibungsphase aktiv ist [OEM99]. Allerdings muss dann die Logik in jedem Reibungsmodell separat enthalten sein. Ein Teil der vorliegenden Arbeit beschäftigt sich mit der Theorie und Numerik der verallgemeinerten Lösung nach Filippov und zeigt die Implementierung für Modelica-Modelle. Damit ist es möglich, eine erweiterte Problemklasse unstetiger multidisziplinärer Modelle in Modelica numerisch zu behandeln.

Die tiefer liegende Problematik, Reibung und Stöße in mechanischen Systemen einheitlich zu modellieren, besitzt eine lange Historie. Ausführliche Literaturlisten sind hierzu

in [Ste00, Glo01] enthalten. Eine verbreitete Methode zur Modellierung dieser Systeme liegt darin, die Folge von differential-algebraischen Gleichungssystemen durch Komplementaritätsprobleme zu ersetzen. Diese Formulierung basiert darauf, zusätzlich zu den Differentialgleichungen die Bedingungen für Kontakte und Stöße so einzubringen, dass für gewisse Variablenpaare stets eine Variable zu Null und die andere nicht negativ wird (Komplementaritätsbedingung). Glocker [Glo01] entwickelte einen mathematischen Rahmen zur Beschreibung der Dynamik von Starrkörpersystemen, in dem Differentialinklusionen (insbesondere Reibungsmodellierung) und unstetige Stoß-Kraftgesetze (Zusammenhänge, um unstetige Geschwindigkeiten zu erzwingen) zu Maß-Differentialinklusionen verknüpft werden. Grob gesagt verhelfen die Maße dazu, die unstetigen Zustandsänderungen direkt in den Differentialgleichungen bzw. -inklusionen zu formulieren.

Numerische Verfahren zur Lösung der mechanischen Probleme teilen sich im Wesentlichen in zwei Gruppen auf. Zuverlässige numerische Verfahren zur Lösung von differential-algebraischen Systemen [Pet82a, Füh88, HW96, Arn98] bewähren sich laut [Glo01] bei wenigen Kontakten zwischen Körpern. Die abschnittsweise Integration der Gleichungen wird dabei wie für die Systeme gewöhnlicher Differentialgleichungen über die Nullstellen von Schaltfunktionen gesteuert. „Time-Stepping“-Algorithmen [Mor88, ST96, För07] beruhen im Gegensatz dazu auf dem Ansatz, Integrale der Kontaktkräfte anstatt der Werte selbst zu betrachten. Man ist damit in der Lage, sowohl die Komplementaritätsbedingungen der Kontakte als auch die Stoßgesetze zu berücksichtigen. Da die genaue Lokalisierung der Kontaktzeitpunkte entfällt, ergeben sich für mechanische Systeme mit vielen Kontaktpunkten effiziente numerische Verfahren.

## Sensitivitätsanalyse und Anwendungen

Die Sensitivitätsanalyse von Differentialgleichungsmodellen hat zum Ziel, Ableitungen (Sensitivitäten) des Lösungsvektors bezüglich Modellparametern zu berechnen. Typische Anwendungen der Sensitivitätsanalyse sind gradientenbezogene Optimierungsverfahren, die z.B. im Rahmen der Parameteridentifizierung Verwendung finden. Im klassischen Fall eines Systems gewöhnlicher Differentialgleichungen mit stetig differenzierbarer rechter Seite erhält man die Sensitivitäten als Lösung der Sensitivitäts-/Variationsgleichungen, für deren numerische Lösung zahlreiche Verfahren zur Verfügung stehen. Dazu wurden in der Vergangenheit bestehende Implementierungen von Zeitintegrationsverfahren erweitert, die zusätzlich zu den ursprünglichen Modellgleichungen die Sensitivitätsgleichungen lösen. Als Beispiele seien Implementierungen von Mehrschrittverfahren in Forschung [LP99, SH03] und kommerziellen Anwendungen [Arn01, Sch04b] erwähnt. Callies [Cal00] entwickelte Runge-Kutta-Verfahren, die der Systemlösung vorseilend Sensitivitäten berechnen.

Für die Auswertung der rechten Seite der Sensitivitätsgleichungen müssen die partiellen Ableitungen der rechten Seite berechnet werden. Hierfür existieren mehrere Möglichkeiten. Analytische Ableitungen können nur für kleinere Probleme zuverlässig aufgestellt werden. Finite Differenzenquotienten sind einfach zu implementierende Näherungen an die Ableitungen, die teilweise unzureichende Genauigkeiten erzielen. Zu einer Funktion, die in einer Programmiersprache als separates Computerprogramm vorliegt, erzeugen automatische Differenzierungsalgorithmen ein neues Programm, das die Ableitungen der Funktion bezüglich der vorgegebenen Variablen berechnet.

Automatisches algorithmisches Differenzieren [Gri00] beruht auf den analytischen Ableitungen elementarer Funktionen in Kombination mit der wiederholten Anwendung der Kettenregel. In den letzten Jahrzehnten wurden für verschiedene Programmiersprachen Algorithmen [BCKP94, BRM96, BBV06] entwickelt, die automatisches Differenzieren von Programmcodes durchführen. Daneben wird in [MKA00] eine Möglichkeit aufgezeigt, mit komplexen statt reellen Rechenoperationen sehr genaue Ableitungsauswertungen zu erzeugen. Dazu werden die Datentypen und Operationen des Programms mit den komplexen Varianten überladen und die Ableitungen aus dem Imaginärteil der zu differenzierenden Funktion approximiert. Der einfacheren Implementierung der „Complex-Step“-Methode stehen bei gleicher Genauigkeit längere Rechenzeiten im Vergleich zum algorithmischen Differenzieren gegenüber [MSA01].

Die komplementäre Methodik zu den Sensitivitätsgleichungen ist in den adjungierten Gleichungen zu sehen. Diese linearen Differentialgleichungen mit einer Systemmatrix, die der negativen transponierten Jacobimatrix der rechten Seite des Modellsystems entspricht, definieren einen Vektor adjungierter Variablen. Mit Hilfe der Variablen lassen sich Ableitungen nach Parametern von zustandsabhängigen Funktionen, die nur an einem bestimmten Zeitpunkt ausgewertet werden, berechnen. Die Besonderheit liegt darin, dass die Werte der adjungierten Variablen zum Endzeitpunkt des gesamten betrachteten Zeitintervalls vorgegeben werden. Damit ist eine rückwärtsgerichtete numerische Zeitintegration nötig, um die adjungierten Gleichungen zu lösen. Hierfür muss vorher die Lösung des eigentlichen Anfangswertproblems numerisch bestimmt werden, damit die transponierte Jacobimatrix entlang der Lösung ausgewertet werden kann. Bock [Boc87] stellt hierzu die Methodik und eine Implementierung für Runge-Kutta-Verfahren zur Verfügung. Neuere Entwicklungen [LP02, SH03, Wal03] setzen BDF-Verfahren für die Lösung der adjungierten Gleichungen ein.

Unstetigkeiten im Differentialgleichungsmodell werfen die Fragestellung auf, wie sich Unstetigkeiten in der rechten Seite und unstetige Zustandsänderungen auf die Existenz der Sensitivitäten auswirken. Im Allgemeinen sind die Sensitivitäten ebenfalls unstetig. Rozenvasser [Roz67] berechnet für Systeme gewöhnlicher Differentialgleichungen mit unstetiger rechter Seite die Sprungbedingungen der Sensitivitäten an den Unstetigkeitsstellen. Für abschnittsweise definierte differential-algebraische Systeme wird in [Fee98] nachgewiesen, dass die Sensitivitäten unter gewissen Regularitätsvoraussetzungen existieren. Formeln zur Berechnung der Sensitivitäten am Anfang der einzelnen Integrationsabschnitte werden angegeben. Die adjungierten Variablen resultieren für unstetige Modelle aus abschnittsweise festgelegten adjungierten Gleichungen. An den Anschlussstellen der Abschnitte erfahren auch die adjungierten Variablen unstetige Änderungen [GM80].

Implementierungen von Sensitivitätsrechnungen mit Unstetigkeiten berücksichtigen die theoretischen Gegebenheiten und berechnen die Anfangswerte der Sensitivitätszustände bzw. der adjungierten Variablen an jeder Nullstelle der Schaltfunktionen neu. In [HP98] werden analytisch aufgestellte Sensitivitätsgleichungen eines abschnittsweise glatten Modells zur elektrischen Stromversorgung mit der Trapezregel integriert. Feehery [Fee98] löst mit dem BDF-Code DSL48S die Sensitivitätsgleichungen und wertet die partiellen Ableitungen für das Residuum der Sensitivitätsgleichungen über automatisch differenzierte Routinen aus. Als Anwendung werden Optimalsteuerungsprobleme mit Ungleichungsbeschränkungen anhand von Beispielen aus der Chemie und Verfahrenstechnik gelöst.

Optimalsteuerung bzw. Optimierung generell stellen das Hauptanwendungsgebiet der Sensitivitätsrechnungen von Differentialgleichungen dar. Unbekannte Parameterwerte eines Modells lassen sich mit Hilfe von Optimierungsverfahren dadurch bestimmen, dass der Unterschied zwischen Simulations- und Messergebnissen minimiert wird. Die dafür verwendeten numerischen Verfahren der restringierten Optimierung, allen voran das SQP-Verfahren, sind auf zuverlässig berechnete Ableitungen der Zielfunktion und der Nebenbedingungen angewiesen.

Bauer [Bau99] identifiziert Parameterwerte mit der Mehrzielmethode und BDF-Verfahren. Für die optimale Versuchsplanung der verfahrenstechnischen Anwendungsbeispiele werden zweite Ableitungen der Zustände nach Parametern durch Integration der entsprechenden Sensitivitätsgleichungen gelöst. Optimalsteuerungsprobleme großer Modelle mit dünn besetzten Ableitungsmatrizen werden in [BP04] mit ähnlichen Techniken untersucht. Unstetigkeiten in den Modellgleichungen werden hierbei mitberücksichtigt. Mit der Parameteridentifizierung eines Eisenbahndrehgestells beschäftigt sich die Arbeit [Gru96], in der mit einer erweiterten Version [Füh88] von DASSL in jedem Zeitschritt die numerischen Lösungen mit variierten Parameterwerten berechnet werden. Finite Differenzenquotienten innerhalb der Integration liefern so die Sensitivitäten für das SQP-Verfahren. Das Paket DASSL [Pet82a] ist ein in der technischen Simulation weit verbreiteter BDF-Code.

Åkeson modelliert in [Åke07] die Trockenpartie einer Papiermaschine mit Modelica. Es werden im Zusammenhang mit Parameteroptimierung und prädiktiver Regelungsverfahren die Sensitivitäten der Modellzustände mit der DASSL-Erweiterung DASPK3.1 [LP99] integriert. Ein Schwerpunkt der Arbeit liegt in der Entwicklung von Optimierungssprachelementen für Modelica-Modelle und der darunter liegenden Software-Werkzeuge. Bisher werden unstetige Modellbeschreibungen nicht unterstützt, was die Anwendung vieler Modelica-Modelle nicht möglich macht. Die vorliegende Arbeit greift diese aktuelle Problemstellung auf und ermöglicht die Sensitivitätsanalyse *unstetiger* Modelica-Modelle in Kombination mit der Parameteridentifizierung komplexer Systeme.

## 1.2 Zielstellung und Aufbau der Arbeit

Mit der Arbeit werden drei wesentliche Ziele verfolgt. Erstens soll die Behandlung von Unstetigkeiten in multidisziplinären Systemmodellen um die Funktionalität erweitert werden, verallgemeinerte Lösungen nach Filippov berechnen zu können. Die Sensitivitätsanalyse der damit vergrößerten Problemklasse unstetiger Systeme definiert das zweite Ziel. Es sollen die Existenz der Sensitivitäten für die typischen Strukturen der multidisziplinär modellierten Systeme geklärt und Berechnungsmethoden vorgeschlagen werden. Die theoretische Seite der verallgemeinerten Lösungen und der Sensitivitätsanalyse soll um die praktische Umsetzung in einer Software-Umgebung ergänzt werden. Implementierungsaspekte, numerische Tests an Modelica-Modellen sowie die Anwendung in der Parameteridentifizierung sind Bestandteile der dritten Zielstellung.

Die Arbeit gliedert sich neben Einleitung und Zusammenfassung in fünf Themenkomplexe. Kapitel 2 gibt eine kompakte Einführung in die einheitliche physikalische Systemmodellierung mit der dafür entwickelten Modellierungssprache Modelica, in der explizit nichtglatte



und diskrete Vorgänge beschrieben werden können. Die Auswirkungen der entsprechenden Sprachelemente auf die Struktur der resultierenden Modellgleichungen werden allgemein dargestellt. Zudem werden wichtige Schritte der symbolischen Modelltransformation erläutert, die alle Modellinformationen in eine Form bringt, um numerische Integrationsverfahren zur zuverlässigen Lösung der Modellgleichungen verwenden zu können.

Liegen die unstetigen Modelle mit Umschaltelementen in Zustandsform vor, so stellt sich die Frage, wie die Lösung dieser Modelle beschrieben werden kann. Kapitel 3 gibt darauf Antworten, indem verschiedene Schaltfälle analysiert und Lösungswege dafür aufgezeigt werden. Wir nehmen die Filippov-Lösung mit in diese Fälle auf, um etwa Reibungsvorgänge einfach modellieren zu können. Spezielle Probleme, die im Zusammenhang mit automatisch erzeugten Modellgleichungen auftreten, führen im zweiten Abschnitt zur Abschwächung gängiger Voraussetzungen an die Schaltfunktion. Damit ist es möglich, eine gewisse Klasse von nichtglatten Schaltfunktionen einzubeziehen, die in natürlicher Weise entstehen. Praxisrelevante Modelle führen zudem zu Schaltvorgängen, die von mehreren Schaltern gleichzeitig ausgelöst werden. Hierfür wird ein Lösungskonzept entwickelt.

Zentraler Bestandteil der Arbeit ist Kapitel 4, das die Sensitivitätsanalyse der nichtglatten Modelle theoretisch aufbereitet. Aufbauend auf den Ergebnissen des vorherigen Kapitels spielen hierfür die Variationsgleichungen glatter gewöhnlicher Differentialgleichungen eine wichtige Rolle. Die Auswirkungen von Modellumschaltungen auf die Existenz von Sensitivitäten teilen sich in die Aspekte während und nach der Umschaltung auf. Selbiges gilt für den alternativen Ansatz über die adjungierten Gleichungen. In dem aufgezeigten einheitlichen Konzept ist bereits die Analyse der Filippov-Lösung enthalten. Die Sensitivitätsanalyse von Modellen mit mehreren gleichzeitigen Schaltvorgängen erfordert einen tiefer gehenden Einblick in die Lösungsstruktur. Dieser wird im zweiten Abschnitt des Kapitels gegeben und an Hand von Beispielen verdeutlicht.

Das anschließende Kapitel 5 trägt die Handschrift der Numerik. Zunächst werden numerische Ansätze dargelegt, die für die Berechnung von Sensitivitäten dynamischer Systeme bekannt sind. Die vorgestellte Implementierung zur Lösung der Sensitivitätsgleichungen verknüpft den BDF-Code DASPK3.1, den automatisch erzeugten C-Code der Modelica-Modelle sowie den neu entwickelten Code FISEMO, der die Erkennung und Berechnung von Filippov-Lösungen sowie die Sensitivitätsanalyse an den Unstetigkeitsstellen der Modellgleichungen übernimmt. Mehrere Beispielmmodelle demonstrieren die Funktionalität der entwickelten Software und zeigen erreichbare Genauigkeiten im Vergleich zu finiten Differenzenapproximationen.

Berechnungen mit einem komplexen Modell eines Industrieroboters zeigen in Kapitel 6 die Anwendbarkeit der implementierten Ansätze. Die Sensitivitätsergebnisse des Modells fließen in einen gradientenbasierten Optimierungsalgorithmus (SQP), der iterativ versucht, die Modellparameter so zu bestimmen, dass das simulierte Verhalten im Modell dem gemessenen Verhalten auf einem Roboterprüfstand möglichst nahe kommt. Detaillierte numerische Tests stellen dazu den Einfluss der Sensitivitätsgenauigkeiten auf den optimalen Zielfunktionswert und die optimalen Parameter heraus.

Im abschließenden Kapitel 7 werden die wesentlichen Ergebnisse der Arbeit zusammengefasst sowie zukünftige Forschungsrichtungen für die Sensitivitätsanalyse automatisch erzeugter Systemmodelle mit Unstetigkeiten vorgeschlagen.

## 2 Unstetige multidisziplinäre Modelle

Die Simulation technischer Systeme trägt wesentlich dazu bei, Entwicklungszeiten für neue Produkte zu verkürzen und aufwendige Tests an Prototypen auf eine minimale Anzahl zu beschränken. Ein gewichtiger Anteil der Modellvorstellungen wird über Differentialgleichungssysteme mathematisch beschrieben. Die zunehmende Integration der technischen Systeme bedingt dabei disziplin-übergreifende Modellierungsansätze, die zu multidisziplinären Gesamtmodellen führen. Ein solches Gesamtmodell besteht aus einem Differentialgleichungssystem, das von den Teilsystemen der einzelnen physikalischen Komponenten aufgebaut wird. In der Praxis ist der Modellaufbau für komplexere Systeme nur mit softwaretechnischen Hilfsmitteln möglich. Häufig ist dafür eine eigene Sprache – die Modellierungssprache – Grundlage. In der Sprache werden die mathematischen Gleichungen des Modells so formuliert, dass klare Schnittstellen zwischen verbundenen Teilmodellen existieren und insgesamt eine automatisierte Erstellung der Gesamtmodelle möglich ist.

Oft enthalten komplexe Modelle zur Vereinfachung schneller Dynamiken unstetige oder nichtglatte Komponenten, die die numerische Lösung der zugehörigen Modellgleichungen mit Standardverfahren erschweren, aber bei richtiger Umsetzung deutlich beschleunigen können. Neben typischen Anwendungen multidisziplinärer Modellierungstechniken stellen wir die Basiselemente der eigens darauf ausgerichteten Modellierungssprache Modelica [Til01, Ott00] vor. Die Beschreibung von Unstetigkeiten und diskreten Effekten wird in einer erweiterten Gleichungsstruktur ausgedrückt, welcher gesonderte Beachtung geschenkt wird. Schließlich beschreiben wir einen aus der Literatur bekannten Weg, wie Modelica-Modelle in eine Form gebracht werden können, die zur zuverlässigen numerischen Lösung der Gesamtmodelle geeignet ist.

### 2.1 Multidisziplinäre Modellierung mit Modelica

Zahlreiche Anwendungen demonstrieren die Notwendigkeit und das Potential multidisziplinärer Modellierung und Simulation [Cel91, Pel03]. Nachfolgend sind Beispiele aus zwei verschiedenen Anwendungsgebieten aufgeführt. Es werden anschließend die Grundzüge von Modelica dargelegt und an einfachen Beispielen veranschaulicht. Zudem beschreiben wir, wie abstrahierte physikalische Gesetzmäßigkeiten, unstetige Vorgänge und diskrete Größen in der Sprache dargestellt werden können.

Die Systeme an Bord eines Verkehrsflugzeugs werden typischerweise vom Triebwerk mit Energie versorgt. Zu den Systemen zählen etwa Flugsteuerung, Fahrwerk, elektrisches Netz und Klimaanlage. Die Energieformen unterscheiden sich je nach System deutlich, so dass in der Summe mechanische, hydraulische, elektrische und pneumatische Leistung vom Triebwerk an die Flugzeugsysteme bereitgestellt werden muss. Will man den

gesamten Energieverbrauch analysieren, dann hilft der multidisziplinäre Modellierungsansatz, die verschiedenen Energiearten in einem Gesamtmodell zu simulieren [BHPS05]. Auf dieser Grundlage kann dann in einem zweiten Schritt der Leistungsbedarf der Systeme modellbasiert optimiert werden [SPB06].

Im Entwurfsprozess von Straßenfahrzeugen ist es nötig, das dynamische Verhalten der Fahrzeuge, insbesondere auch Wechselwirkungen von Antriebssträngen mit Fahrdynamik und Komfortanforderungen zu prüfen. Die Beurteilung des Energieverbrauchs bzw. der Wirkungsgrade des Gesamtfahrzeugs stellt eine weitere wichtige Grundlage des rechnergestützten Fahrzeugentwurfs dar. Mit multidisziplinärer Modellierung lassen sich aktuelle und zukünftige Fahrzeugarchitekturen, z. B. ein Allradantrieb mit aktiven Differentialen oder ein parallel hybrider elektrischer Antrieb mit Rückspeisung von Bremsenergie, auf die genannten Kriterien hin untersuchen [TOB07]. Konzepte zur Umwandlung der Abgaswärme eines Verbrennungsmotors in elektrische Energie werden in [EUT06] multidisziplinär simuliert.

### 2.1.1 Die Modellierungssprache Modelica

In diesem Abschnitt werden wesentliche Ideen und Konzepte der Modellierungssprache Modelica aufgezeigt. Die Sprache ermöglicht eine gleichberechtigte Beschreibung von Systemen aus verschiedenen physikalischen Teilgebieten. Die Ausführungen beziehen sich auf [Fri04, Til01] sowie die Artikelfolge [Ott00]. In Modelica werden die zu modellierenden Systeme ihrem physikalischen Aufbau entsprechend objektorientiert strukturiert. Einzelne Komponenten eines Systems werden als eigene Objekte definiert und stehen über Verbindungslinien, die physikalische Verbindungen repräsentieren, mit anderen Komponenten in Interaktion.

In Abbildung 2.1 ist ein Objektdiagramm eines einfachen Modells dargestellt, das einen Elektromotor beschreibt. Es besteht aus den Komponenten elektrischer Widerstand, elektrische Spule, elektromotorische Kraft (EMF) und mechanische Trägheit, die über Linien miteinander verbunden sind. Blaue Linien kennzeichnen dabei eine elektrische Verbindung, graue Linien stellen eine starre mechanische Verbindung her. Die beiden linken

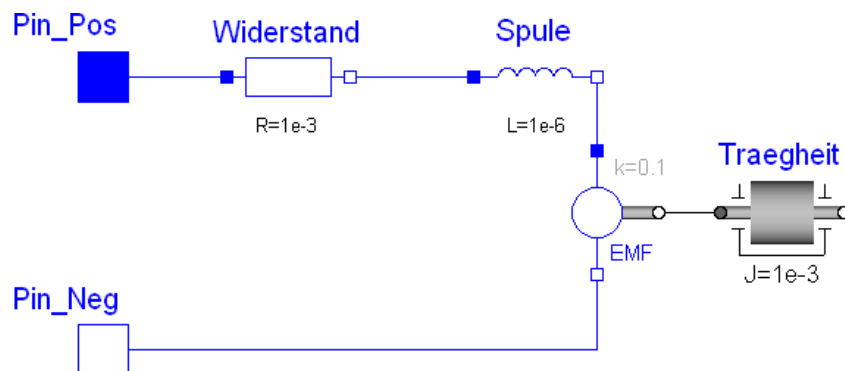


Abbildung 2.1: Modelica-Objektdiagramm für ein Motormodell.



Objekte (`Pin_Pos`, `Pin_Neg`) dienen als Schnittstellen für den positiven und negativen elektrischen Anschluss des Motors (z. B. an eine Spannungsquelle). Jede einzelne Komponente kann wiederum für sich aus mehreren verschalteten Teilkomponenten bestehen, die auf dieser Ebene aber nicht sichtbar sind. Die hierarchische Strukturierung trägt wesentlich zur Übersichtlichkeit der Modelldiagramme bei. Die Schnittstellen jeder Komponente bilden das Verbindungsglied zwischen dem Inneren und dem Äußeren der Komponente. Deswegen gibt es auch eine eigene Modellklasse `connector`, mit der die verschiedenen physikalischen Schnittstellen definiert werden können.

Eine Schnittstellenklasse kann aus mehreren Variablen bestehen, die jeweils zu einer der beiden Kategorien *Fluss* und *Potential* gehören. Flussvariablen, wie der elektrische Strom, beschreiben Größen, deren Werte sich in einem Verbindungsknoten zu Null addieren (Kirchhoff'sche Knotenregel). Potentialvariablen bleiben dagegen in verbundenen Knoten gleich (Kirchhoff'sche Maschenregel). Ein Beispiel hierfür ist das elektrische Potential. Das Prinzip der Fluss- und Potentialvariablen lässt sich auf vielen physikalischen Teilgebieten anwenden. So sind in diesem Sinn Kraft, Drehmoment, Volumenstrom, Wärmefluss und Massenstrom Flussvariablen sowie Weg, Winkelposition, spezifische Enthalpie und Druck Potentialvariablen.

Auf der untersten hierarchischen Ebene einer Komponente besteht jede Teilkomponente aus Gleichungen, die den Zusammenhang zwischen den zeitabhängigen Schnittstellengrößen, konstanten Komponentenparametern und der Zeit herstellen. Insbesondere können diese Gleichungen Zeitableitungen von Variablen zur Beschreibung von dynamischen Vorgängen enthalten, die über den `der`-Operator definiert werden. Alle Gleichungen sind in Modelica deklarativ. Somit findet keine Zuweisung von Ausdrücken in einer vorgegebenen Reihenfolge statt. Die Gleichungen werden als Identitäten im mathematischen Sinn verstanden, so dass es keinen Unterschied macht, welche Ausdrücke links oder rechts vom Gleichheitszeichen stehen. Im nachfolgenden Beispiel 2.1 ist das Modelica-Codefragment der linearen elektrischen Spule notiert, deren Strom  $i$  mit der anliegenden Spannung  $v$  über die Differentialgleichung  $L \frac{di}{dt} = v$  gekoppelt ist.

### Beispiel 2.1 (Modelica-Modellcode einer linearen elektrischen Spule)

<pre>connector PositivePin   SI.Voltage v;   flow SI.Current i; end PositivePin;  connector NegativePin   SI.Voltage v;   flow SI.Current i; end NegativePin;</pre>	<pre>partial model OnePort   SI.Voltage v;   SI.Current i;   PositivePin p;   NegativePin n; equation   v = p.v - n.v;   0 = p.i + n.i;   i = p.i; end OnePort;</pre>	<pre>model Inductor   extends OnePort;   parameter     SI.Inductance L=1; equation   L*der(i) = v; end Inductor;</pre>
---	---	--

□

An diesem Beispiel wird mit der Vererbung ein weiterer wichtiger Bestandteil von Modelica deutlich. Das Teilmodell `OnePort`, das zwei elektrische Anschlüsse (`PositivePin`, `NegativePin`) als Schnittstellen enthält, wird über das Schlüsselwort `extends` an das

Spulenmodell **Inductor** vererbt. Es kann aber auch an andere Modelle, etwa an das Widerstandsmodell vererbt werden, so dass die grundlegende Struktur einer elektrischen Komponente mit zwei Anschlüssen nur einmalig definiert werden muss.

Jedes Komponentenmodell kann Parameter enthalten, die bezüglich der Zeit konstant sind. Beispielsweise werden so der Wert der Induktivität  $L$  oder des linearen Widerstands  $R$  festgelegt. Die Werte können von darüber liegenden Modellhierarchien geändert werden. Über ähnliche Konstrukte können sogar ganze Modellklassen ersetzt werden. Details sind in den angegebenen Literaturstellen bzw. in der Modelica-Sprachspezifikation [Mod07] zu finden.

Ein grundlegendes Problem in der Systemmodellierung besteht darin, eine angemessene Modellierungstiefe zu definieren oder zu finden. Werden physikalische Effekte berücksichtigt, die nur sehr geringen Einfluss auf die zu untersuchenden Eigenschaften des Systems haben, so kann die numerische Integration der Modellgleichungen schnell rechenintensiv werden und zu einem schlechten Verhältnis zwischen Rechenaufwand und Modellzweck führen. Andererseits liefern Modelle mit approximierten oder vernachlässigten physikalischen Zusammenhängen evtl. unbefriedigende Ergebnisse im Vergleich zum tatsächlichen, beobachteten Verhalten des Systems. Diese Problematik ergibt sich insbesondere für multidisziplinäre Modelle, wenn mehrere Komponenten aus verschiedenen physikalischen Anwendungsgebieten (etwa Elektronik und Mechanik) zusammen beschrieben werden.

Daher ist es üblich und nützlich, gewisse Vorgänge zur Vereinfachung der Modelle und zur Beschleunigung der Zeitintegration als unstetig anzunehmen. Eine Beschleunigung erreicht man im Allgemeinen nur, wenn die modellierten Unstetigkeiten während der Integration einer gesonderten Behandlung unterzogen werden. In der klassischen Literatur [SW95, DB02] wird für den Einsatz der numerischen Integrationsverfahren vorausgesetzt, dass die Modellgleichungen hinreichend glatt sind. Wird diese Voraussetzung verletzt, kann es zu numerischen Problemen kommen, die beispielsweise die Schrittweitensteuerung ineffizient arbeiten lassen [HNW93]. Um eine einheitliche, sachgerechte und automatisierbare Behandlung der unstetigen Vorgänge zu gewährleisten, ist deshalb in Modelica die Beschreibung von Unstetigkeiten von Grund auf vorgesehen. Im folgenden Abschnitt gehen wir näher auf dieses Konzept ein.

### 2.1.2 Unstetige Modellierungselemente in Modelica

Modellunstetigkeiten und strukturvariable Systeme treten in einer Vielzahl von Anwendungen auf: Reibungsphänomene oder Impulse in der Mechanik, Schaltungsvorgänge in der Elektrotechnik sowie analoger und digitaler Reglerentwurf [ESF02, UGS99]. Als Unstetigkeiten wollen wir Vorgänge verstehen, die nicht hinreichend glatt modelliert sind, um die numerische Lösung zuverlässig und effizient berechnen zu können. Die am häufigsten verwendete Methode, Unstetigkeiten in Modelica zu formulieren, ist die bedingte Verzweigung mittels **if-then-else**-Konstrukten. Sie ermöglichen die Modellierung des Komponentenverhaltens in verschiedenen Arbeitspunkten bzw. -bereichen. Der Anspruch von Modelica liegt darin, dass auch verzweigte Gleichungen deklarativ bleiben [Ott00]. Somit ist nicht von vornherein festgelegt, welche Größen gegeben und welche gesucht sind. Als illustrierendes Beispiel soll eine ideale elektrische Diode dienen.

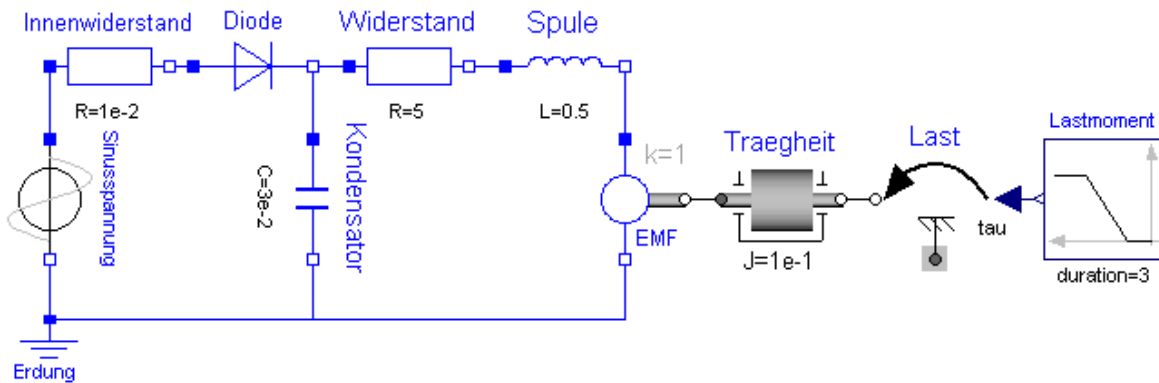


Abbildung 2.2: Modelica-Objektdiagramm für das erweiterte Motormodell mit Spannungsquelle und Lastmoment.

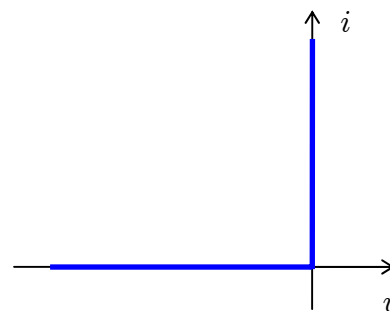
### Beispiel 2.2 (Ideale Diode)

Die als ideal angenommene Diode besitzt die abstrahierte Strom-Spannungs-Charakteristik aus nachstehendem Diagramm. Die Diode fungiert damit als idealer Schalter, der für negative anliegende Spannungen öffnet und für positive Ströme schließt. Nach [OREM00] lässt sich eine derartige Charakteristik über eine parametrisierte Kurve mit einem Kurvenparameter  $s$  wie folgt in Modelica beschreiben:

```

model IdealDiode
  extends OnePort;
  Boolean off;
  Real s;
equation
  off = (s < 0);
  v = if off then s else 0;
  i = if off then 0 else s;
end IdealDiode;

```



Wir erkennen das unstetige Verhalten, wenn  $s$  gleich Null wird, da dann evtl. andere Äste der bedingten Verzweigungen aktiv werden. Unstetigkeiten treten hier möglicherweise für Zeitableitungen von  $v$  oder  $i$  auf. Zudem wird sichtbar, dass die Modellgleichungen deklarativ sind, da beispielsweise die Variable  $s$  im Modell zunächst nicht bekannt ist und erst durch äußere Vorgaben an  $v$  bzw.  $i$  bestimmt werden kann.  $\square$

Als Anwendung des Diodenmodells ist in Abbildung 2.2 eine Erweiterung des Motormodells aus Abbildung 2.1 zu sehen. Zur Spannungsversorgung des Motors wird eine sinusförmige Wechselspannungsquelle mit Innenwiderstand an einen einfachen Gleichrichter, bestehend aus einer idealen Diode und einem Glättungskondensator, angeschlossen. Die mechanische Last des Motors wird über ein zeitabhängiges Momentenprofil vorgegeben. Das Gesamtmodell ist unstetig, da es unstetige Modellkomponenten enthält. An Beispiel 2.2 wird zudem ersichtlich, dass neben Variablen des Datentyps `Real` auch boolesche Variablen (hier: `off`) definiert werden können. Boolesche Variablen gehören wie Integer-

Variablen zu den *diskreten* Variablen, die ihre Werte nur beim Zutreffen einer Bedingung ändern können. Es lassen sich auch diskrete Real-Variablen definieren, um mit deren Hilfe etwa digitale Regelgesetze zu reproduzieren [Ott00, Teil 7]. Zur Beschreibung solcher Situationen ist eine zusätzliche Anweisung (**when** *Bedingung* **then**) nötig, welche Gleichungen kennzeichnet, die nur an diskreten Zeitpunkten gelten sollen. Im Gegensatz zu **if-then-else**-Blöcken ist ein **when**-Block nur dann aktiv, wenn die Bedingung wahr *wird*, aber nicht mehr, wenn sie zeitlich anschließend wahr bleibt. Damit können Änderungen an diskreten Variablen innerhalb von **when**-Blöcken vorgenommen werden. Auf die bzgl. der Zeit linksseitigen Grenzwerte der diskreten Variablen kann mit dem Operator **pre** zugegriffen werden.

### Beispiel 2.3 (Diskretes lineares Zustandsraummodell als Modelica-Code)

Analog zu [Fri04, Chapter 13.3.1.4] wird ein diskretes lineares Zustandsraummodell mit Abtastzeit  $T$  definiert. Zu den Zeitpunkten  $t_i := i \cdot T$  ( $i = 1, 2, 3, \dots$ ) werden für die diskreten Variablen  $x$  und  $y$  neue Werte gemäß der Rekursion des Zustandsraummodells berechnet:

$$\begin{aligned} x(t_i) &= x(t_{i-1}) + 2u(t_i), & x(0) &= 0, \\ y(t_i) &= 3x(t_{i-1}) + 4u(t_i), & y(0) &= 0. \end{aligned}$$

Das entsprechende Modelica-Modell lautet:

<pre> block StateSpace   parameter Real a = 1, b = 2;   parameter Real c = 3, d = 4;   parameter Real T = 0.1;   input Real u;   discrete output Real y;   discrete Real x;   ... </pre>	<pre> ... equation   when sample(0, T) then     x = a*pre(x) + b*u;     y = c*pre(x) + d*u;   end when; end StateSpace; </pre>
--	--

□

**When**-Blöcke dienen nicht nur zur Änderung von diskreten Variablen, sondern lassen auch unstetige Änderungen von Variablen zu, für die Zeitableitungen (**der**-Operator) ins Modell eingehen. Mit der Funktion **reinit** können die Werte solcher Variablen geändert werden. Dieses Vorgehen ist z. B. bei der Modellierung von Impulsen hilfreich, wie Beispiel 2.4 zeigt.

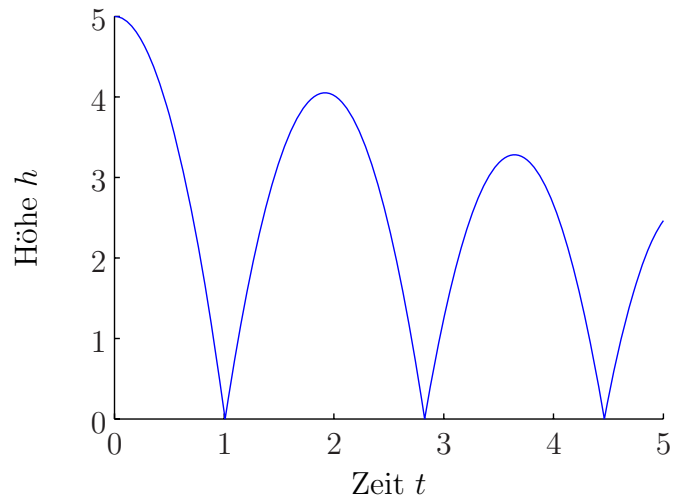
### Beispiel 2.4 (Springender Ball [Ott01])

Ein kleiner Ball befindet sich in der Nähe der Erdoberfläche im freien Fall ( $\ddot{h} = -9.81$ ), bis er den Boden berührt ( $h = 0$ ). Beim Aufprall erfährt die Geschwindigkeit  $v$  eine unstetige Änderung in die entgegengesetzte Richtung, da elastische Verformungen vernachlässigt werden. Energieverluste im Stoß werden durch die Stoßzahl  $e$  mit  $0 < e < 1$  modelliert. Der Ball fällt von der Anfangshöhe  $h = 5$  auf den Boden und erreicht in der folgenden Zeit immer niedrigere Maximalhöhen gemäß nebenstehendem Verlauf. In Modelica liest sich das Modell wie folgt:

```

model BouncingBall
  parameter Real e = 0.9;
  parameter Real g = 9.81;
  Real h, v;
  initial equation
    h = 5;
  equation
    der(h) = v;
    der(v) = -g;
    when h <= 0 then
      reinit(v, -e*v);
    end when;
end BouncingBall;

```



□

## 2.2 Umformung der Modellgleichungen

Wir wollen uns einen Überblick verschaffen, welche Gleichungsstrukturen sich aus den oben vorgestellten Sprachelementen von Modelica ergeben. Dazu lösen wir uns von der objektorientierten Darstellung des Modells, um das Gesamtsystem analysieren zu können. Dieses System drückt dann die Struktur der Gleichungen aus, eignet sich aber im Allgemeinen nur bedingt zur numerischen Lösung. Daher werden grundsätzliche symbolische Transformationen beschrieben, die auf die Gleichungen angewendet werden können, um deren Struktur zu verbessern [Ott00]. Für den kontinuierlichen Teil der Gleichungen ergibt sich als Resultat die (numerische) Zustandsform des differential-algebraischen Systems mit Schaltfunktionen.

### 2.2.1 Struktur der Gleichungen

Um die Gesamtheit der Gleichungen für ein vorhandenes Modell zu erzeugen, werden alle Vererbungen der Objekte in den Teilmodellen explizit durchgeführt und entsprechende Variablen und Gleichungen eingefügt. Die Verbindungstopologie wird durch die entsprechenden Gleichungen der Verbindungsvariablen ersetzt. Dabei ist die Summe der Flussvariablen in jedem Verbindungsknoten Null, verbundene Potentialvariablen sind gleich zu setzen, siehe auch Abschnitt 2.1.1.

Die Gleichungsstruktur eines allgemeinen Modelica-Modells wird in der Literatur nicht einheitlich wiedergegeben. Die Darstellungen in [Mod07, Appendix C] und [Fri04, BAF06] unterscheiden sich zum Teil im Detaillierungsgrad und zum Teil in den Argumentlisten der verschiedenen Funktionen. Wir wählen die detaillierteste Form aus [BAF06] und ergänzen diese um die Gleichungen zu Beginn des Integrationsintervalls. Des Weiteren stellen wir die diskreten Gleichungen  $F^m = 0$  in impliziter Form dar, da diese Form nach unserer

Tabelle 2.1: Variablenbezeichnungen und Funktionen eines allgemeinen Modelica-Modells aus Gleichung (2.1).

Bez.	Erläuterung
$w$	Vektor aller Variablen, auf die der <b>der</b> -Operator angewendet wird.
$w^R$	Vektor der Komponenten von $w$ , für die <b>reinit</b> -Anweisungen vorhanden sind.
$\dot{w}$	<b>der</b> ( $w$ ).
$u$	Vektor aller auf oberster Modellebene als Eingabegrößen ( <b>input</b> ) definierten Variablen.
$z$	Vektor aller restlichen kontinuierlichen (d. h. nicht diskreten) Variablen.
$t$	Die unabhängige Größe Zeit.
$m$	Vektor aller diskreten Variablen des Modells.
$m_{\text{pre}}$	<b>pre</b> ( $m$ ).
$p$	Vektor aller Konstanten und Parameter.
$c$	Vektor von zusätzlichen (nicht im Modell definierten) booleschen Variablen, die die Werte der Bedingungen (bei <b>if</b> - und <b>when</b> -Konstrukten, auch in booleschen Gleichungen) enthalten.
$v$	Gesamtvektor aller Größen eines Modells: $(\dot{w}, w, z, u, t, m, m_{\text{pre}}, p)$ .
$v^B$	Vektor der Komponenten von $v$ , die boolesch sind.
Funktion	Erläuterung
$F^A$	Allen <b>initial</b> -Gleichungen und Parameterabhängigkeiten entsprechend.
$F^{\dot{w},z}$	Allen kontinuierlichen Gleichungen entsprechend.
$F^m$	Allen diskreten Gleichungen (in und außerhalb von <b>when</b> -Blöcken) entsprechend.
$F^c$	Allen booleschen Verknüpfungen in Bedingungen und logischen Ausdrücken (nicht innerhalb von <b>when</b> -Blöcken) entsprechend.
$R$	Vektor aller Relationen (nicht innerhalb von <b>when</b> -Blöcken).
$F^w$	Alle Ausdrücke in <b>reinit</b> -Anweisungen.

Auffassung besser den Möglichkeiten der Modellierungssprache entspricht. Das Gesamtgleichungssystem

$$F^A(v) = 0, \quad (2.1a)$$

$$F^{\dot{w},z}(v, c) = 0, \quad (2.1b)$$

$$F^m(v, c) = 0, \quad (2.1c)$$

$$F^c(v^B, R(v)) = c, \quad (2.1d)$$

$$F^w(v, c) = w^R \quad (2.1e)$$

ist die Grundlage für die im nächsten Abschnitt definierten hybriden Systeme in numerischer Zustandsform. In Tabelle 2.1 sind alle vorkommenden Bezeichnungen für Variablen

und Funktionen erläutert. Es wird zwischen diskreten Größen  $m$  und  $c$  sowie kontinuierlichen Variablen  $w$ ,  $z$  und  $t$  unterschieden. Mischformen können in den Größen  $u$  und  $p$  auftreten. Die Funktionen  $F^A$ ,  $F^{\dot{w},z}$  und  $F^m$  ergeben sich aus einer einfachen symbolischen Umformung aller entsprechenden Modellgleichungen. Die rechte Seite aller zugehörigen Gleichungen wird durch Subtraktion (für boolesche Variablen mit den Werten 0/1) in die linke Seite mit aufgenommen, damit die rechte Seite der Gleichungen (2.1b) und (2.1c) verschwindet. Im Vektor  $R$  werden alle Relationen des Modells (außer innerhalb von **when**-Blöcken) zusammengefasst, um in der Funktion  $F^c$  die im Modell vorkommenden Bedingungen und logischen Ausdrücke darstellen zu können. Die kontinuierliche Gleichung (2.1b) wird durch Gleichung (2.1e) mit der Funktion  $F^w$  ergänzt, in der alle Ausdrücke enthalten sind, die in **reinit**-Anweisungen vorkommen. Nachfolgend sind für die Beispiele aus Abschnitt 2.1.2 die Modellgleichungen in Form (2.1) angegeben:

### Beispiel 2.5 (Allgemeine Gleichungsstruktur für Beispiel 2.2)

Die Modellgleichungen der idealen Diode stellen zwar kein Gesamtmodell dar, das für sich alleine lösbar wäre. Dennoch geben wir die Gleichungen des Modells mit den Bezeichnungen des Systems (2.1) an:

$$z := (\text{p.v}, \text{n.v}, \text{p.i}, \text{n.i}, \text{v}, \text{i}, \text{s})^\top, \quad m = m_1 := \text{off}, \quad F^m(v, c) := m_1 - c_1, \quad R(v) := (z_7 < 0),$$

$$F^{\dot{w},z}(v, c) := \begin{pmatrix} z_5 - z_1 + z_2 \\ -z_3 - z_4 \\ z_6 - z_3 \\ z_5 - (\text{if } c_2 \text{ then } z_7 \text{ else } 0) \\ z_6 - (\text{if } c_2 \text{ then } 0 \text{ else } z_7) \end{pmatrix}, \quad F^c(v^B, r) := \begin{pmatrix} r \\ m_1 \end{pmatrix}.$$

□

### Beispiel 2.6 (Allgemeine Gleichungsstruktur für Beispiel 2.3)

$$u = u_1 := \text{u}, \quad m := (\text{x}, \text{y}, \text{T\_sample})^\top, \quad p := (\text{a}, \text{b}, \text{c}, \text{d}, \text{T})^\top, \quad F^A(v) := m, \quad R(v) := (t > m_3),$$

$$F^m(v, c) := \begin{pmatrix} m_1 - (\text{when } c_1 \text{ then } p_1 \cdot m_{1,\text{pre}} + p_2 \cdot u_1) \\ m_2 - (\text{when } c_1 \text{ then } p_3 \cdot m_1 + p_4 \cdot u_1) \\ m_3 - (\text{when } c_1 \text{ then } m_{3,\text{pre}} + p_5) \end{pmatrix}, \quad F^c(v^B, r) := r.$$

□

### Beispiel 2.7 (Allgemeine Gleichungsstruktur für Beispiel 2.4)

$$w := (\text{h}, \text{v})^\top, \quad p := (\text{g}, \text{e})^\top, \quad F^A(v, c) := \begin{pmatrix} w_1 - 5 \\ w_2 \end{pmatrix}, \quad F^{\dot{w},z}(v, c) := \begin{pmatrix} \dot{w}_1 - w_2 \\ \dot{w}_2 + p_1 \end{pmatrix},$$

$$R(v) := (w_1 \leq 0), \quad F^c(v^B, r) := r, \quad F^w(v, c) := (\text{when } c_1 \text{ then } -p_2 \cdot w_2), \quad w^R := w_2. \quad \square$$

Bezüglich der Zeit besteht Gleichungssystem (2.1) im Allgemeinen aus einer Mischform von diskreten und kontinuierlichen Gleichungen. Diskrete Variablen sind auf genügend kleinen Zeitintervallen stets konstant. Wenn man in einer solchen Lösungsphase ist, reicht es aus, für konstante diskrete Variablen nur das Gleichungssystem (2.1b) des kontinuierlichen Teils zu betrachten. Diese Gleichungen bilden dann ein System differentialalgebraischer Gleichungen (englisch: Differential Algebraic Equation, DAE) zur Bestimmung der kontinuierlichen Variablen  $w$ ,  $\dot{w}$  und  $z$ . Im nächsten Abschnitt wird dargelegt, wie DAE-Systeme symbolisch aufbereitet werden können, damit eine effiziente numerische Lösung möglich wird.



## 2.2.2 Numerische Zustandsform der Modellgleichungen

Bekanntermaßen [Pet82b] kann die numerische Lösung von DAE-Systemen größere Probleme bereiten als die Lösung gewöhnlicher Differentialgleichungen. Als grobes Maß für die in der numerischen Lösung zu erwartenden Schwierigkeiten gilt der *Differentiationsindex* (kurz: Index). DAE-Systeme mit einem Index, der größer als Eins ist, erfordern spezielle Techniken, die die Struktur der Gleichungen ausnutzen [HW96, Arn98]. Beispiele dafür sind die Gear-Gupta-Leimkuhler-Formulierung [GGL85] für mechanische Systeme oder die quasilinearen Index-2-Systeme für die elektrische Schaltkreissimulation [Tis96].

Die DAE-Systeme (2.1b), die sich direkt aus der komponentenorientierten Modellierung mit Modelica ergeben, haben typischerweise einen hohen Differentiationsindex ( $\geq 2$ ). Das liegt unter anderem daran, dass in den Verbindungsgleichungen oft Größen gleich gesetzt und damit *Zwangsbedingungen* erzeugt werden. Zwangsbedingungen sind rein algebraische Gleichungen, deren Existenz gerade den Unterschied zwischen Systemen gewöhnlicher Differentialgleichungen und differential-algebraischen Systemen ausmacht. Bei DAE-Systemen müssen zusätzlich zu den gegebenen Zwangsbedingungen oft auch deren Ableitungen („versteckte“ Zwangsbedingungen) berücksichtigt werden. Für die Bewegungsgleichungen mechanischer Mehrkörpersysteme erhält man z. B. aus den Zwangsbedingungen der Lagekoordinaten die Geschwindigkeits-Zwangsbedingungen, die die Lösung ebenfalls erfüllen muss [Füh88]. Insbesondere können die Anfangswerte der Zustände nicht beliebig frei gewählt werden, sondern müssen sowohl die gegebenen als auch die versteckten Zwangsbedingungen einhalten. Die Anfangswerte werden dann als *konsistent* bezeichnet.

Standard-Verfahren wie der Pantelides-Algorithmus [Pan88] und die Methode der „Dummy Derivatives“ [MS93] sorgen bei Modelica-Modellen [MOE00, Ott00] zusammen mit symbolischen Formelmanipulationen dafür, dass alle notwendigen Ableitungen der zu differenzierenden Zwangsbedingungen berechnet werden. Als Resultat erhält man ein System, das dieselbe Lösung wie (2.1b), aber einen niedrigeren Differentiationsindex besitzt. Die übergeordnete Technik, den Index eines DAE-Systems zu reduzieren, heißt daher *Indexreduktion*. Da in vielen Anwendungen in den Komponenten- und Verbindungsgleichungen der Modelle meist nur wenige Variablen auftreten, ist es sinnvoll, die Gleichungen und Variablen des Gesamtsystems (2.1) mit dem Algorithmus von Tarjan [Tar72] so umzusortieren, dass nach möglichst vielen Unbekannten rekursiv aufgelöst werden kann. Zusätzlich verkleinert man die Dimensionen der einzelnen, nach der Sortierung noch verbleibenden impliziten Gleichungsblöcke, indem gewisse Variablen durch ihre zuweisenden Ausdrücke substituiert werden. Im Zusammenhang mit Modelica-Modellen wird diese Vorgehensweise *Tearing* genannt [EO94].

Bezeichnet man die Zustände mit  $x \in \mathbb{R}^{n_x}$  und die sich ergebenden algebraischen Variablen mit  $y \in \mathbb{R}^{n_y}$ , so kann das transformierte System (2.1b) lokal um einen Punkt  $(x, t)$  wie folgt geschrieben werden:

$$\dot{x} = f(x, t), \quad (2.2a)$$

$$y = g(x, t). \quad (2.2b)$$



Die Funktionen  $f : \mathbb{R}^{n_x+1} \rightarrow \mathbb{R}^{n_x}$  und  $g : \mathbb{R}^{n_x+1} \rightarrow \mathbb{R}^{n_y}$  seien dabei ausreichend glatt. Es wird vorausgesetzt, dass alle diskreten Variablen sowie die Parameter  $p$  konstant sind und in der Definition von  $f$  und  $g$  berücksichtigt werden. Ebenso seien die Abhängigkeiten bzgl. der kontinuierlichen Komponenten von  $u$  enthalten. Falls keine algebraischen Variablen auftreten ( $n_y = 0$ ), dann hat das System (2.2) den Differentiationsindex Null, ansonsten Eins. Um die gesuchten kontinuierlichen Variablen mit numerischen Integrationsverfahren berechnen zu können, ist in beiden Fällen die Struktur (2.2) günstiger als die ursprüngliche Form mit evtl. höherem Index. Wir wollen das System (2.2) *numerische Zustandsform* nennen, auch wenn es im Allgemeinen den Differentiationsindex Eins besitzt. Aus Sicht der numerischen Integration ist es aber nur notwendig, das System gewöhnlicher Differentialgleichungen (2.2a) zu lösen. Die Größe  $y$  kann im Nachhinein aus Gleichung (2.2b) berechnet werden, wenn  $x$  ermittelt ist [MOE00].

Das System (2.2) bleibt zeitlich nur solange mit unveränderten Funktionen  $f$  und  $g$  gültig, bis eine Komponente von  $c$  ihren Wert ändert. Wenn dieser Fall eintritt, spricht man oft von einem *Ereignis* (englisch: event). Um feststellen zu können, wann potentielle Ereignisse stattfinden, ist ein Standardansatz, *Schaltfunktionen*  $q : \mathbb{R}^{n_x+1} \rightarrow \mathbb{R}$  einzuführen [HS80, Boc87, HNW93]. Wenn eine Schaltfunktion ihr Vorzeichen wechselt, wird ein Ereignis ausgelöst. Die Schaltfunktionen können in Modelica mit Hilfe der Relationen  $R$  einfach konstruiert werden, indem entsprechende Ausdrücke der rechten Seite der Relation durch Subtraktion auf die linke Seite transformiert werden. Beispielsweise hat  $q$  für die Relation  $R(v) = (t > m_3)$  aus Beispiel 2.6 die Gestalt  $q(x, t) = t - m_3$ . In Bezug auf die fehlenden restlichen Variablen im Argument von  $q$  gilt die analoge Aussage wie für die Funktionen  $f$  und  $g$  in (2.2).

Wegen der veränderten Wahrheitswerte von  $c$  ist es an einem Ereignispunkt notwendig, alle diskreten Variablen zusammen mit allen kontinuierlichen neu zu berechnen. Diese Ereignis-Neuinitialisierung ist insbesondere auch am Anfang der numerischen Integration durchzuführen. Bereits für einfache Beispiele sind die diskreten und die kontinuierlichen Gleichungen so gekoppelt, dass eine direkte Auflösung nach den unbekannten diskreten Variablen nicht möglich ist. Das Modell aus Beispiel 2.2 demonstriert diesen Sachverhalt. Wenn das Diodenmodell in ein Gesamtmodell eingebettet ist, gibt es noch eine Gleichung außerhalb des Modells, die einen Zusammenhang zwischen dem Strom  $i$  und der anliegenden Spannung  $v$  herstellt. Zusammen mit dieser Gleichung lassen sich die Gleichungen des Diodenmodells dadurch lösen, dass eine Annahme für den Wert von `off` gemacht wird. Der dann aus dem restlichen kontinuierlichen Gleichungssystem berechnete Wert für  $s$  dient anschließend zur Überprüfung der Annahme, ob `off` richtig gewählt wurde. Dies motiviert die folgende Iteration [Fri04, Chapter 18.2.5.3] zur Lösung der gemischten diskret-kontinuierlichen Gleichungssysteme:

### Iteration 2.8 (Neuinitialisierung am Ereignispunkt)

Setze Startwerte für die diskreten Variablen.

Wiederhole

1. Löse die kontinuierlichen Gleichungen mit fixierten diskreten Variablen.
2. Berechne die diskreten Variablen aus den kontinuierlichen.

Bis sich die diskreten Variablen beim Übergang von 1. zu 2. nicht mehr ändern.  $\square$

Da oft mehrere Relationen in einem Modell vorhanden sind, gibt es auch mehrere Schaltfunktionen. Sie werden alle parallel betrachtet und lösen unter Umständen zahlreiche Ereignisse während der Lösungsphase aus. Wir setzen voraus, dass das Integrationsintervall dadurch in eine *endliche* Anzahl von Teilintervallen aufgeteilt wird. In den Teilintervallen nimmt das Gesamtsystem jeweils die Form (2.2) mit unterschiedlichen Funktionen  $f$  und  $g$  an. An den Ereignispunkten bestimmen Übergangsbedingungen die Werte der Zustände. Diese Bedingungen resultieren aus den Gleichungen (2.1e) bzw. (2.1a) zusammen mit den Gleichungen für die konsistenten Anfangswerte des zu Grunde liegenden DAE-Systems.

In Definition 2.9 wird der Wechsel zwischen zwei Systemen in numerischer Zustandsform an einem Ereignispunkt mit den entsprechenden Übergangsbedingungen beschrieben. Der Wechsel kann beispielsweise dazu führen, dass neue Zustandsvariablen zum Zustandsvektor hinzu kommen oder bestehende entfallen. DAE-Systeme, die ihre Struktur ereignisbedingt ändern, werden in der Literatur [Bar00, HM05] als *hybride* DAE-Systeme bezeichnet. Wir verwenden diesen Begriff daher auch für die Systeme in numerischer Zustandsform. Die Folge von abschnittsweise definierten glatten Systemen (2.2) verdeutlicht in Bemerkung 2.11 den qualitativen Aufbau eines *mehrschichtigen* hybriden Systems, in das sich ein allgemeines Modelica-Modell transformieren lässt.

### Definition 2.9 (Hybrides System in numerischer Zustandsform)

Es seien zwei Systeme gewöhnlicher Differentialgleichungen

$$\dot{x}^0 = f^0(x^0, t), \quad \dot{x}^1 = f^1(x^1, t)$$

mit stetig differenzierbaren Abbildungen  $f^0 : \mathbb{R}^{n_{x^0}+1} \rightarrow \mathbb{R}^{n_{x^0}}$  und  $f^1 : \mathbb{R}^{n_{x^1}+1} \rightarrow \mathbb{R}^{n_{x^1}}$  gegeben. Der Übergang zwischen beiden Systemen sei durch  $A^1 : \mathbb{R}^{n_{x^0}+1} \rightarrow \mathbb{R}^{n_{x^1}}$  vorgegeben. Zur Zeit  $t^0$  habe  $x^0$  den Anfangswert  $A^0 \in \mathbb{R}^{n_{x^0}}$ . Algebraische Größen  $y^0, y^1$  seien durch die Funktionen  $g^0 : \mathbb{R}^{n_{x^0}+1} \rightarrow \mathbb{R}^{n_{y^0}}$  und  $g^1 : \mathbb{R}^{n_{x^1}+1} \rightarrow \mathbb{R}^{n_{y^1}}$  definiert. Dann heißt

$$\dot{x}^0 = f^0(x^0, t), \quad y^0 = g^0(x^0, t) \quad \text{für} \quad t^0 \leq t \leq t^1, \quad x^0(t^0) = A^0, \quad (2.3a)$$

$$\dot{x}^1 = f^1(x^1, t), \quad y^1 = g^1(x^1, t) \quad \text{für} \quad t^1 \leq t \leq t^\infty, \quad x^1(t^1) = A^1(x^0(t^1), t^1) \quad (2.3b)$$

*hybrides System in numerischer Zustandsform.* Die Schaltzeit  $t^1 > t^0$  sei dabei eindeutig durch die Bedingung

$$q^1(x^0(t^1), t^1) = 0 \quad (2.4)$$

bestimmt, wobei  $q^1 : \mathbb{R}^{n_{x^0}+1} \rightarrow \mathbb{R}$  eine gegebene Schaltfunktion sei und  $x^0$  lokal für  $t > t^1$  existiere. Der Zeitpunkt  $t^\infty > t^1$  sei ein fest gewählter Endpunkt des betrachteten Zeitintervalls.  $\square$

### Bemerkung 2.10 (Eindeutige Lösung eines hybriden Systems)

Wenn für ein hybrides System vorausgesetzt wird, dass die Schaltzeit  $t^1$  eindeutig bestimmbar ist, so folgt sofort die Existenz einer eindeutigen Lösung des hybriden Systems. Will man die eindeutige Bestimmbarkeit von  $t^1$  an Bedingungen knüpfen, so ist mit einer stetig differenzierbaren Schaltfunktion  $q$  dafür hinreichend:

$$Dq^1 := q_{x^0}^1(x^0(t^1), t^1)f^0(x^0(t^1), t^1) + q_t^1(x^0(t^1), t^1) \neq 0. \quad (2.5)$$

Die Aussage folgt aus dem Satz über implizite Funktionen zur Auflösung von Gleichung (2.4) nach  $t^1$ .  $\square$

**Bemerkung 2.11 (Mehrschichtiges hybrides System)**

Im zeitlichen Anschluss an das erste Ereignis kann ein zweites Ereignis zur Zeit  $t^2 > t^1$  eintreten. Allgemein können wir rekursiv eine endliche Folge von Ereignissen für  $i = 1, \dots, m$  mit dazugehörigen Systemen in numerischer Zustandsform definieren:

$$\begin{aligned} \dot{x}^0 &= f^0(x^0, t), & y^0 &= g^0(x^0, t) & \text{für } t^0 \leq t \leq t^1, & & x^0(t^0) &= A^0, \\ \dot{x}^i &= f^i(x^i, t), & y^i &= g^i(x^i, t) & \text{für } t^i \leq t \leq t^{i+1}, & & x^i(t^i) &= A^i(x^{i-1}(t^i), t^i). \end{aligned}$$

Die stetig differenzierbare Schaltfunktion  $q^i : \mathbb{R}^{n_{x^{i-1}}+1} \rightarrow \mathbb{R}$  definiert hierbei eindeutig die Schaltzeit  $t^i > t^{i-1}$ , falls die Bedingung  $Dq^i \neq 0$  analog zu (2.5) erfüllt ist:

$$q^i(x^{i-1}(t^i), t^i) = 0.$$

Der Endpunkt des gesamten Zeitintervalls sei  $t^{m+1} := t^\infty$ . Dieses Gesamtsystem nennen wir *mehrschichtiges* oder  $(m+1)$ -*schichtiges hybrides System*.  $\square$

Die in diesem Abschnitt beschriebene Vorgehensweise eignet sich besonders für objekt-orientiert erstellte Modelle, da sie für eine große Klasse von hybriden DAE-Systemen mit höherem Index eine automatische, rechnergestützte Transformation auf numerische Zustandsform ermöglicht. Zur Modellierung eines multidisziplinären Systems ist es demnach meistens nicht nötig, im Modell spezielle Komponentenverschaltungen zu vermeiden, die einen hohen Index des Gesamtsystems verursachen.

Die Software Dymola [Dym] bietet eine grafische Oberfläche zur multidisziplinären Modellierung mit Modelica und zur numerischen Lösung der Modellgleichungen. Die vorgestellten Techniken, hybride DAE-Systeme numerisch lösbar zu machen, sind darin umgesetzt. Das transformierte Modell wird dabei in C-Code übersetzt und steht anschließend für die numerische Integration zur Verfügung. Die in den Kapiteln 5 und 6 aufgezeigten numerischen Tests basieren alle auf dem von Dymola erzeugten Code. Der Code enthält Routinen zur Neuinitialisierung an Ereignispunkten sowie das entsprechende Modell in numerischer Zustandsform mit Schaltfunktionen.

In den folgenden Kapiteln wird insbesondere die Lösung und die Abhängigkeit der Lösung von den Modellparametern  $p$  untersucht. Da die Modelle nach der Transformation in der Form (2.2) vorliegen, reicht es im Folgenden aus, die hybriden Systeme in numerischer Zustandsform zu studieren.

## 3 Lösungstheorie unstetiger Systeme

Setzt sich ein Modell aus glatten Teilmodellen zusammen, so ist das Gesamtmodell ebenfalls glatt. Wir verstehen unter einem *glatten* Modell die Beschreibung der physikalischen Zusammenhänge mit ausreichend oft stetig differenzierbaren Funktionen. Die Dynamik des Gesamtmodells ist somit über ein System gewöhnlicher Differentialgleichungen  $\dot{x} = f(x, t)$  mit glatter rechter Seite  $f$  darstellbar. Für gegebene Anfangswerte der Zustände gibt es nach dem Satz von Picard-Lindelöf [Kön97] eine (eindeutige) lokale Lösung des dynamischen Systems. Diese verläuft bis an den Rand des Definitionsbereichs der rechten Seite. Analysiert man das Lösungsverhalten hybrider Systeme, dann sind weiterführende Überlegungen insbesondere am Ereignispunkt nötig, da die klassische Theorie gewöhnlicher Differentialgleichungen nicht mehr anwendbar ist.

Im ersten Abschnitt dieses Kapitels stellen wir dazu eine Analyse der qualitativ möglichen Lösungsverläufe nach einem Ereignis vor. Hier verwenden wir einen erweiterten Lösungsbegriff (Filippov-Lösung), der es ermöglicht auch Systeme zu behandeln, für die keine klassische Lösungen definiert sind. Darauf folgend sind im zweiten Abschnitt Problemklassen thematisiert, die bei praktischen, automatisch erzeugten Modellen auftreten, die aber die üblichen Standardvoraussetzungen nicht mehr erfüllen. Abgeschwächte Voraussetzungen garantieren in dem vorgestellten Lösungskonzept dennoch eindeutige Lösungen.

### 3.1 Schaltsituationen am Ereignispunkt

Die Unstetigkeiten in multidisziplinären Modellen aus Kapitel 2 werden durch Ereignisse angezeigt, die über Nullstellen der Schaltfunktionen ausgelöst werden. Jede Schaltfunktion selbst hat ihren Ursprung in einer Relation des Modells, die stets den logischen Wert wahr oder falsch besitzt und entsprechende Zweige in den Modellgleichungen bedingt. Der Wahrheitswert der Relation wirkt wie ein Schalter auf die Zweige, somit ist die Assoziation zwischen den zwei möglichen Schalterstellungen und dem Vorzeichen der zugehörigen Schaltfunktion nahe liegend. Hat die Schaltfunktion den Wert Null, ist die Schalterstellung für die an das Ereignis anschließende Zeit nicht offensichtlich.

#### 3.1.1 Klassische Lösung

Für diesen und die folgenden Abschnitte ist es ausreichend, Systeme mit unstetiger rechter Seite gemäß nachfolgender Definition als Problemklasse zu untersuchen. Die hybriden Systeme (2.3) führen wir in Bemerkung 3.8 unter gewissen Voraussetzungen auf Systeme mit unstetiger rechter Seite zurück.

Tabelle 3.1: Lösungssituationen am Schaltzeitpunkt  $t^*$  mit  $q(x(t^*), t^*) = 0$ .

Fall	$D^+q(x^*, t^*)$	$D^-q(x^*, t^*)$	Lokal für $t > t^*$ : $q(x(t), t)$
1	$> 0$	$> 0$	$> 0$
2	$< 0$	$< 0$	$< 0$
3	$> 0$	$< 0$	$\neq 0$
4	$< 0$	$> 0$	$= 0$

**Definition 3.1 (System mit unstetiger rechter Seite)**

Gegeben sei das System gewöhnlicher Differentialgleichungen

$$\dot{x} = \begin{cases} f^+(x, t), & \text{falls } q(x, t) > 0, \\ f^-(x, t), & \text{falls } q(x, t) < 0 \end{cases} \quad (3.1)$$

mit glatten rechten Seiten  $f^+, f^- : \mathbb{R}^{n_x+1} \rightarrow \mathbb{R}^{n_x}$ . Es seien zusätzlich die Anfangswerte  $x^* := x(t^*) = A \in \mathbb{R}^{n_x}$  zu einem Zeitpunkt  $t^*$  und eine stetige Schaltfunktion  $q : \mathbb{R}^{n_x+1} \rightarrow \mathbb{R}$  bekannt, die über ihr Vorzeichen die rechte Seite der Differentialgleichung festlegt. Diese Gleichung nennen wir *System mit unstetiger rechter Seite* oder kurz *unstetiges System*.  $\square$

In Definition 2.9 wird vorausgesetzt, dass bekannt ist, mit welcher rechten Seite die Lösung nach dem Ereignis fortgesetzt wird. Wir analysieren nun, wie sich entscheiden lässt, welcher Zweig der rechten Seite zu wählen ist. Richtungsableitungen der Schaltfunktion spielen hierbei eine entscheidende Rolle.

**Definition 3.2 (Richtungsableitungen der Schaltfunktion)**

Für ein unstetiges System (3.1) mit glatter Schaltfunktion definieren wir die *Richtungsableitungen*  $D^+q$ ,  $D^-q$  der Schaltfunktion  $q$  in die Richtung möglicher Lösungen:

$$D^+q := q_x f^+ + q_t, \quad D^-q := q_x f^- + q_t. \quad \square$$

Gilt für eine glatte Schaltfunktion  $q(x^*, t^*) = 0$ , so geben die Vorzeichen der Richtungsableitungen  $D^+q(x^*, t^*)$ ,  $D^-q(x^*, t^*)$  Auskunft über die Fortsetzbarkeit der Lösung  $x$ , da sie das qualitative Verhalten der Schaltfunktion  $q$  entlang der verschiedenen Lösungskurven wiedergeben, die entweder von  $f^+$  oder  $f^-$  erzeugt werden. Tabelle 3.1 verschafft einen Überblick, welche wesentlichen Fälle auftreten können [Boc87, Eic92, Cal00]. Die letzte Spalte für das Vorzeichen von  $q$  ist als Folge der Vorzeichen von  $D^+q$  und  $D^-q$  zu verstehen.

**Definition 3.3 (Schaltfläche)**

Für ein unstetiges System (3.1) heißt die Menge

$$\Sigma := \{(x, t) : q(x, t) = 0\}$$

*Schaltfläche* im erweiterten Phasenraum  $\mathbb{R}^{n_x+1}$ .  $\square$

In den ersten beiden Fällen von Tabelle 3.1 kann die Lösung nur mit der aufgezeigten Schalterstellung für die rechte Seite fortgesetzt werden, da die jeweils andere Schalterstellung im Widerspruch zur Richtungsableitung von  $q$  steht. Die Lösung durchdringt die Schaltfläche  $\Sigma$  im Punkt  $(x^*, t^*)$ , weil die rechte Seite am Ereignispunkt auf den anderen Zweig umschaltet. Da die Richtungsableitungen nicht verschwinden, schneidet die Lösung die Schaltfläche in einem Winkel, der ungleich Null ist. Die Fälle, in denen  $D^+q(x^*, t^*)$  oder  $D^-q(x^*, t^*)$  verschwinden, werden separat weiter unten behandelt (siehe Tabelle 3.2).

Der dritte Fall in Tabelle 3.1 würde eine Lösung für jede der beiden Schalterstellungen ermöglichen. Diese Situation ist für praktisch relevante Beispiele aber nicht zutreffend und spielt daher nur eine theoretische Rolle. Deswegen wird der Fall nicht weiter untersucht. Fall 4 ist dann gegeben, wenn sich die Schaltfunktion genau entgegengesetzt zu dem verhält, wie es für die verschiedenen Schalterstellungen nötig wäre, damit sie gültig bleiben könnten. Diese Situation des *inkonsistenten Schaltens* lässt keine (klassische) Lösung zu. Ein erweitertes Lösungskonzept wird hierfür im nächsten Abschnitt besprochen.

### 3.1.2 Filippov-Lösung

Abbildung 3.1 verdeutlicht die Situation für Fall 4 aus Tabelle 3.1. Die erweiterten Zustandsableitungen  $(f^+, 1)$  und  $(f^-, 1)$  zeigen lokal um den Punkt  $(x^*, t^*)$  im erweiterten Phasenraum in die Schaltfläche  $\Sigma$  hinein. Die Lösung kann sich also nicht von der Schaltfläche entfernen – sie muss in ihr bleiben. Filippov [Fil88] schlägt für solche Fälle vor, den klassischen Lösungsbegriff zu erweitern und mengenwertige rechte Seiten  $\Psi(x, t)$  einzubeziehen. Dieser Ansatz führt zu Differentialinklusionen  $\dot{x} \in \Psi(x, t)$ , für die es eine umfassende Theorie (etwa in [AC84]) gibt. Allgemein besteht  $\Psi(x, t)$  nur aus einem Element, solange die rechte Seite stetig ist. Für die Definition von  $\Psi$  an einer Unstetigkeitsstelle gibt es verschiedene Vorgehensweisen. Diese hängt von der physikalischen Gesetzmäßigkeit ab, die durch die unstetige rechte Seite beschrieben wird. Wir wollen uns in der Arbeit auf die konvexe Hülle  $\Psi_F(x, t) := \text{conv} \{f^+, f^-\}$  als mengenwertige rechte Seite beschränken. Dieses Konzept geht auf Filippov zurück und deckt wichtige Anwendungsfälle ab. In Beispiel 3.11 wird ein anderer Ansatz kurz erläutert.

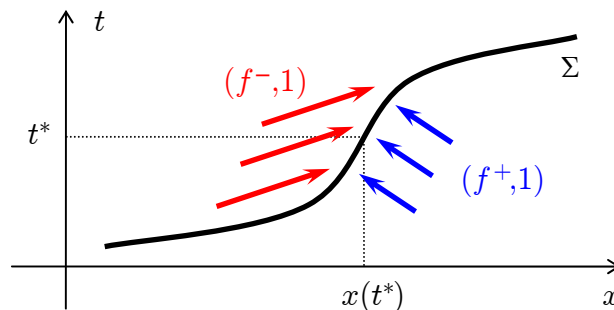


Abbildung 3.1: Zustandsableitungen in der Nähe der Schaltfläche  $\Sigma$  im Fall einer Filippov-Lösung.

Tabelle 3.2: Lösungssituationen am Schaltzeitpunkt  $t^*$  mit  $q(x(t^*), t^*) = 0$  und verschwindenden Richtungsableitungen.

Fall	$D^+q(x^*, t^*)$	$D^-q(x^*, t^*)$	Lokal für $t > t^*$ : $q(x(t), t)$
5	$> 0$	$= 0$	$\neq 0$
6	$= 0$	$< 0$	$\neq 0$
7	$< 0$	$= 0$	$\leq 0$
8	$= 0$	$> 0$	$\geq 0$
9	$= 0$	$= 0$	$\in \mathbb{R}$

Mit der Forderung, dass die Lösung in der Schaltfläche verlaufen soll, wird aus  $\Psi$  die gesuchte rechte Seite ausgewählt. Für  $\Psi_F$  bedeutet dies:

$$\dot{x} = \lambda f^+ + (1 - \lambda)f^- \quad \text{mit } \lambda \in [0, 1]$$

und  $q(x(t), t) = 0$  für  $t \geq t^*$ . Dieses DAE-System mit Differentiationsindex Zwei kann durch Differentiation der algebraischen Gleichung  $q(x, t) = 0$  explizit in das System

$$\dot{x} = \frac{D^+q f^- - D^-q f^+}{D^+q - D^-q}$$

überführt werden [HNW93]. Zusammenfassend erhalten wir folgende Definition:

#### Definition 3.4 (Filippov-Lösung)

Es sei vorausgesetzt, dass für ein unstetiges System (3.1) mit glatter Schaltfunktion die folgenden Zusammenhänge gelten:  $q(x^*, t^*) = 0$ ,  $D^+q(x^*, t^*) < 0$  und  $D^-q(x^*, t^*) > 0$ . Dann heißt die Lösung  $x$ , die folgendes System gewöhnlicher Differentialgleichungen lokal für  $t \geq t^*$  erfüllt, *Filippov-Lösung*:

$$\dot{x} = f^*(x, t) := \frac{D^+q f^- - D^-q f^+}{D^+q - D^-q}(x, t).$$

Die Filippov-Lösung wird oft auch als *Sliding Motion* [Fil88, §4] bezeichnet. □

#### Bemerkung 3.5 (Eigenschaften der Filippov-Lösung)

Die Filippov-Lösung existiert zeitlich für  $t > t^*$  mindestens solange bis  $D^+q$  oder  $D^-q$  einen Nulldurchgang aufweisen, vorausgesetzt die Lösung bleibt beschränkt. Aus Stetigkeitsgründen ist entlang einer Filippov-Lösung stets  $D^+q \leq 0$  und  $D^-q \geq 0$ . Deshalb kann man die Richtungsableitungen als neue Schaltfunktionen betrachten [Boc87], die ein Ereignis zu einem späteren Zeitpunkt  $t > t^*$  definieren, wenn die Lösung die Schaltfläche wieder verlässt. Die bisherige Schaltfunktion  $q$  wird bis zu diesem Ereignis nicht als Schaltfunktion benutzt, da die Filippov-Lösung garantiert, dass  $(x(t), t) \in \Sigma$  und damit  $q(x(t), t) = 0$  gilt. □



Die Situation eines späteren Ereignisses mit Nullstellen von  $D^+q$  oder  $D^-q$  lässt sich in die allgemeine Problematik, welche Fälle für verschwindende Richtungsableitungen existieren, eingliedern. Tabelle 3.2 listet als Fortsetzung von Tabelle 3.1 alle prinzipiellen Möglichkeiten auf. Wir nehmen dabei an, dass eine Filippov-Lösung in der Schaltfläche nur dann von Interesse ist, wenn es keine klassische Lösung mit  $f = f^+$  oder  $f = f^-$  gibt. Da in den Fällen 5 und 6 klassische Lösungen existieren, ist hier die Filippov-Lösung lokal für  $t > t^*$  nicht relevant. Wegen der Stetigkeit von  $D^+q$  (Fall 5) bzw.  $D^-q$  (Fall 6) treten diese Fälle nicht während oder am Ende einer Phase mit Filippov-Lösung auf. In Beispiel 3.6 werden für alle Fälle Beispiele angegeben, die demonstrieren, dass die Lösungssituationen, wie sie in der letzten Spalte von Tabelle 3.2 notiert sind, auch existieren.

### Beispiel 3.6 (Verschwindende Richtungsableitungen)

Es seien die Schaltfunktion  $q(x, t) := x$ , der Zeitpunkt  $t^* = 0$  und der Anfangswert  $A = 0$  zu einem skalaren System mit verschiedenen rechten Seiten  $f^+$  und  $f^-$  gegeben:

Beispiel	$f^+(x, t)$	$f^-(x, t)$	Korrespondierender Fall	Rechte Seite lokal für $t > t^*$
A.a	1	$t^2$	5	$f^+$
A.b	1	$-t^2$	5	$f^+, f^-$
B.a	-1	$t^2$	7	$f^* = 0$
B.b	-1	$-t^2$	7	$f^-$
C.a	$-t^2$	$t^2$	9	$f^* = 0$
C.b	$\frac{1}{2}t^2$	$t^2$	9	$f^+$
C.c	$-\frac{1}{2}t^2$	$-t^2$	9	$f^-$
C.d	$t^2$	0	9	$f^+, f^-$

Für alle Beispielvarianten lässt sich in Abbildung 3.2 erkennen, welche Möglichkeiten es für die Wahl von  $f$  gibt. Diese sind in der obigen Tabelle in der letzten Spalte eingetragen. In den Fällen (A.b) und (C.d) gibt es keine eindeutige rechte Seite für  $t > t^*$ , die Situation ähnelt der aus Fall 3 in Tabelle 3.1. Es ist offensichtlich, dass für die Beispiele (A.b), (C.d) die Lösung nur durch Setzen der Anfangszustände in die Schaltfläche gelangt. Für die restlichen Beispiele gibt es Lösungen, die in positiver Zeitrichtung in die Schaltfläche hineinlaufen oder sie schneiden. Beispiele zu den korrespondierenden Fällen 6 und 8 (siehe Tabelle 3.2) erhält man einfach aus den Beispielen (A.a), (A.b), (B.a) und (B.b), indem man den Übergang  $q \rightarrow -q$  mit dem entsprechenden Vertauschen von  $f^+$  und  $f^-$  vornimmt.  $\square$

### Bemerkung 3.7 (Fortsetzung der Lösung in nicht eindeutigen Fällen)

Im Fall 5 (Fall 6 analog) aus Tabelle 3.2 existiert eine Lösung für  $f = f^+$ . Das Beispiel 3.6 (A.b) zeigt, dass es in manchen Situationen möglich ist, zusätzlich eine Lösung mit  $f = f^-$  zu konstruieren. Wenn wir annehmen, dass es stets eine eindeutige Lösung des unstetigen Systems gibt, so scheidet diese Situation aus. In den Fällen ist dann klar, dass  $f^+$  die neue rechte Seite für  $t > t^*$  ist. Die Fälle 7, 8 und evtl. 9 treten im Zusammenhang mit Filippov-Lösungen auf, wenn eine Richtungsableitung eine Nullstelle hat, siehe Bemerkung 3.5. Lässt man keine mehrfachen Nullstellen von  $D^+q$ ,  $D^-q$  zu, so verlässt



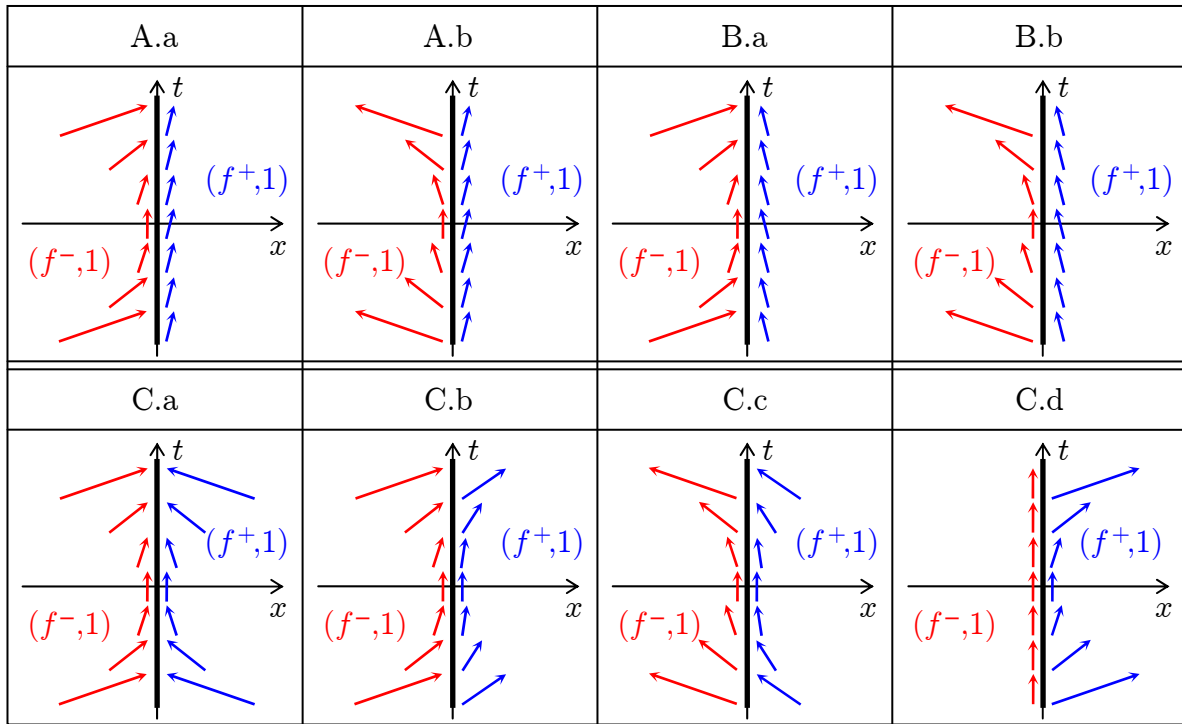


Abbildung 3.2: Zustandsableitungen in der Nähe der Schaltfläche  $\Sigma = \{(x, t) : x = 0\}$  für alle Fälle aus Beispiel 3.6.

die Lösung die Schaltfläche mit einer klassischen Lösung. Im Fall 9 ist zunächst nicht klar, ob die Lösung mit  $f = f^+$  oder mit  $f = f^-$  fortgesetzt wird. In allen Fällen, in denen  $D^+q$  oder  $D^-q$  verschwinden und in denen die analytischen Lösungen nicht bekannt sind, können evtl. höhere Ableitungen von  $q(x(t), t)$  bei entsprechender Glattheit eine genauere Analyse ermöglichen. Auch hier ist das Ziel, festzustellen, wie sich die Schaltfunktion qualitativ für die verschiedenen Lösungsäste verhält.  $\square$

Bisher hatten wir in diesem Kapitel Systeme mit unstetiger rechter Seite betrachtet. Den Zusammenhang mit hybriden Systemen (2.3), die zur unstetigen rechten Seite zusätzlich Übergangsbedingungen in den Zuständen enthalten können, klären wir in der folgenden Bemerkung.

### Bemerkung 3.8 (Einbettung in hybride Systeme)

Für ein hybrides System (2.3) definieren wir  $x := x^1$ ,  $t^* := t^1$ ,  $A := A^1(x^0(t^1), t^1)$ . Wir setzen nun voraus, dass eine glatte Schaltfunktion  $\rho^1$  mit  $\rho^1(x^1(t^1), t^1) = 0$  existiert, die also eine Nullstelle für die geänderten Zustände am Ereignispunkt besitzt. Mit dieser sei  $q := \rho^1$ . Es sei darauf hingewiesen, dass die von  $x^0$  abhängende Schaltfunktion  $q^1$  des hybriden Systems unter Umständen nur dazu verwendet wird, den Ereigniszeitpunkt  $t^1$  festzulegen. Auf welche rechte Seite für die Zeit  $t > t^1$  umgeschaltet wird, richtet sich dann nach  $\rho^1$ . Im Spezialfall  $A^1 = \text{id}_{x^0}$  gilt:  $q = \rho^1 = q^1$ . Mögliche rechte Seiten für die  $x^1$  bestimmende Differentialgleichung seien je nach Vorzeichen von  $\rho^1$  als  $f^+$  und  $f^-$

Tabelle 3.3: Parameter und Variablen für Beispiel 3.9, Abbildung 2.2.

Bez.	Modelica-Variable	Beschreibung	Wert	Einheit
$R_i$	Innenwiderstand.R	Innenwiderstand	$10^{-2}$	$\Omega$
$C$	Kondensator.C	Kapazität des Kondensators	$3 \cdot 10^{-2}$	F
$R$	Widerstand.R	Widerstand des Motors	5	$\Omega$
$L$	Spule.L	Induktivität der Spule	$\frac{1}{2}$	H
$k$	EMF.k	EMF-Konstante	1	Nm/A
$J$	Traegheit.J	Trägheitsmoment	$10^{-1}$	kg m <sup>2</sup>
$v_C$	Kondensator.v	Spannung am Kondensator	$v_C(t)$	V
$i_L$	Spule.i	Strom durch die Spule	$i_L(t)$	A
$\omega$	Traegheit.w	Winkelgeschwindigkeit	$\omega(t)$	rad/s
$v_G$	Sinussspannung.v	Versorgungsspannung	$325 \sin(10\pi t)$	V
$M_L$	Last.tau	Lastdrehmoment	$M_L(t)$	Nm

bezeichnet. Algebraische Größen  $y$  spielen für die Situationen am Ereignispunkt keine Rolle und stellen insofern keine Einschränkung dar.

Mit diesen Definitionen liegt ein System (3.1) mit unstetiger rechter Seite vor und alle Aussagen dieses Kapitels gelten auch für hybride Systeme, nachdem die Zustandsänderungen am Ereignispunkt ausgeführt sind. Hat keine Schaltfunktion mit den geänderten Zuständen eine Nullstelle, so sind alle Schalterstellungen über die Vorzeichen der Schaltfunktionen eindeutig festgelegt, und eine weitere Analyse am Ereignispunkt ist nicht notwendig. Die einzige Einschränkung, die wir hier vornehmen, ist die Verfügbarkeit der die Zustandsänderung beschreibenden Funktion  $A^1$ , bevor über die korrekte rechte Seite entschieden wird. Für komplizierte Modelle kann man sich eine Kopplung der Zustandsänderung mit der rechten Seite vorstellen. In der Mechanik gibt es Verfahren, die eine konsistente Kombination von Kontaktfällen [Kli97, Glo01] über die Lösung eines linearen Komplementaritätsproblems ermitteln.  $\square$

### 3.1.3 Beispiele unstetiger Systeme

Die nachfolgenden Beispiele beleuchten unterschiedliche Aspekte der gezeigten Ansätze zur Lösung von Systemen mit unstetiger rechter Seite. Beispiel 3.9 zeigt den klassischen Durchgang der Lösung durch die Schaltfläche, wohingegen die Beispiele 3.10 und 3.11 Filippov-Lösungen besitzen. Letzteres gibt einen kurzen Einblick in ein weiteres Lösungskonzept neben der Filippov-Lösung.

#### Beispiel 3.9 (Belasteter Elektromotor I)

Das vereinfachte Modell eines belasteten (ungeregelten) Elektromotors aus Abbildung 2.2 soll die klassische Lösung eines unstetigen Systems verdeutlichen. Mit den Bezeichnungen aus Tabelle 3.3 und einer vorgegebenen, glatten Funktion  $M_L$  für das Lastdrehmoment

ist das Gesamtmodell durch das abschnittsweise lineare System

$$\frac{d}{dt} \begin{pmatrix} v_C \\ i_L \\ \omega \end{pmatrix} = \begin{pmatrix} a_{11} & -\frac{1}{C} & 0 \\ \frac{1}{L} & -\frac{R}{L} & -\frac{k}{L} \\ 0 & \frac{k}{J} & 0 \end{pmatrix} \begin{pmatrix} v_C \\ i_L \\ \omega \end{pmatrix} + \begin{pmatrix} b_1(t) \\ 0 \\ \frac{1}{J}M_L(t) \end{pmatrix}$$

darstellbar. Es werden dabei die abkürzenden Bezeichnungen

$$(a_{11}, b_1(t)) := \begin{cases} \frac{1}{R_i C}(-1, v_G(t)), & \text{falls } v_G(t) - v_C > 0, \\ (0, 0), & \text{falls } v_G(t) - v_C < 0 \end{cases}$$

verwendet. Für die Anfangswerte der Zustände  $x := (v_C, i_L, \omega)^\top$  gelte:  $x(0) = 0$ . Die Schaltfunktion  $q(x, t) := v_G(t) - v_C$  definiert über ihre Nullstellen Ereignispunkte  $(x^*, t^*)$ . An einem solchen gilt:

$$\begin{aligned} D^+q(x^*, t^*) &= \frac{1}{C}i_L^* + \dot{v}_G(t^*) + \frac{1}{R_i C}(v_C^* - v_G(t^*)), \\ D^-q(x^*, t^*) &= \frac{1}{C}i_L^* + \dot{v}_G(t^*). \end{aligned}$$

Da  $q(x^*, t^*) = v_G(t^*) - v_C^*$  verschwindet, folgt die Identität  $D^+q(x^*, t^*) = D^-q(x^*, t^*)$ . Damit erhalten wir Fall 1 oder 2 aus Tabelle 3.1 für  $\dot{v}_G(t^*) \neq -\frac{1}{C}i_L^*$ . Aus Stetigkeitsgründen haben  $D^+q$  bzw.  $D^-q$  lokal für  $t < t^*$  dieselben Vorzeichen wie am Ereignispunkt. Folglich durchstößt die Lösungskurve  $(x(t), t)$  die Schaltfläche  $\Sigma$  im Punkt  $(x^*, t^*)$ . Die numerische Lösung des Modells mit einem sprunghaften Lastmoment ist in Abbildung 5.2 von Kapitel 5 visualisiert.  $\square$

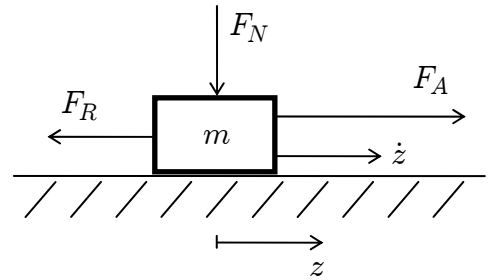
### Beispiel 3.10 (Coulomb-Reibung nach [ESF02])

Die Modellierung von Reibungsvorgängen ist ein wichtiger Bestandteil der dynamischen Beschreibung mechanischer Systeme. Als einfaches Modellsystem betrachten wir einen Quader der Masse  $m$ , der sich über eine ebene Fläche in  $z$ -Richtung bewegt. Das *Coulomb-Modell* definiert den Betrag der maximal wirkenden Reibungskraft  $F_R$  als

$$|F_R| = \mu|F_N|$$

mit einem Reibungskoeffizienten  $\mu \geq 0$  und der Normalkraft  $F_N$  senkrecht zur Kontaktfläche. Dabei ist die Normalkraft  $F_N = mg$  mit der Erdbeschleunigung  $g$  gegeben. Die Richtung der Reibungskraft ist entgegengesetzt zur Geschwindigkeit  $\dot{z}$  der Masse. Wir nehmen an, dass zudem eine zeitabhängige Anregungskraft  $F_A$  auf die Masse wirkt. Man erhält mit den Zuständen  $x = (z, x_2)^\top$  eine unstetige rechte Seite der Bewegungsgleichung:

$$\dot{x} = f(x, t) = \begin{cases} f^+(x, t) := \begin{pmatrix} x_2 \\ \frac{1}{m}F_A(t) - \mu g \end{pmatrix}, & \text{falls } q(x, t) := x_2 > 0, \\ f^-(x, t) := \begin{pmatrix} x_2 \\ \frac{1}{m}F_A(t) + \mu g \end{pmatrix}, & \text{falls } q(x, t) < 0. \end{cases}$$



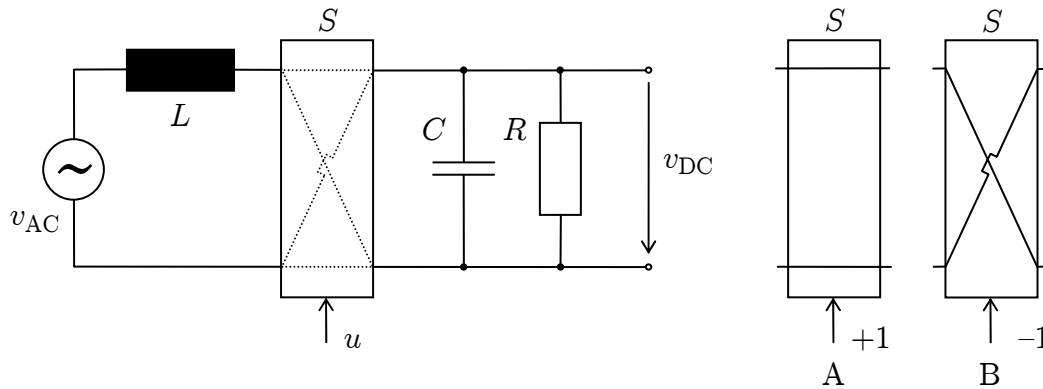


Abbildung 3.3: Links: Schaltbild eines AC/DC-Konverters. Rechts: Die beiden möglichen Schaltzustände A und B der Schaltlogik  $S$ .

Mit den Werten  $m = 1$ ,  $g = 10$ ,  $\mu = \frac{1}{2}$ ,  $F_A(t) = 8 \sin 2\pi t$ ,  $z(0) = 0$ ,  $\dot{z}(0) = 0$  ergibt sich gemäß Definition 3.2 für  $t^* = 0$ :

$$D^+q(x(0), 0) = f_2^+(x(0), 0) = -5 < 0, \quad D^-q(x(0), 0) = f_2^-(x(0), 0) = 5 > 0.$$

Nach den theoretischen Überlegungen aus Abschnitt 3.1.2 definiert man für  $t > 0$  über die rechte Seite  $f := f^* = 0$  die Filippov-Lösung, bis in diesem Fall  $D^+q$  eine Nullstelle besitzt. Physikalisch erhält man genau das gewünschte Verhalten des Modells: Die Masse bleibt solange in Ruhe, bis die Anregungskraft größer als die maximale Reibungskraft wird.  $\square$

### Beispiel 3.11 (AC/DC-Konverter [UGS99])

Ein elektrischer Konverter von Wechsel- auf Gleichspannung sei durch die linke Schaltung in Abbildung 3.3 beschrieben. Ein Wechselspannungsgenerator mit der Frequenz  $\omega$  erzeugt eine Spannung  $v_{AC} = V_{AC} \sin \omega t$ , die über eine Induktivität  $L$  an eine Schaltlogik  $S$  übertragen wird. Ein Steuersignal  $u$ , das die Werte  $+1$  oder  $-1$  annehmen kann, legt fest, ob die Verschaltung nach Schema A oder B erfolgt. Die resultierende Ausgangsspannung  $v_{DC}$  soll einem vorgegebenen Spannungsverlauf  $v_{Soll}$  folgen.

Ein einfacher Sliding-Mode-Regler setzt

$$u = \begin{cases} +1, & \text{falls } q(x, t) := (v_{Soll} - v_{DC})i_{AC} > 0, \\ -1, & \text{falls } q(x, t) < 0. \end{cases}$$

Damit erhält man mit  $x := (i_{AC}, v_{DC})^T$  das System

$$\dot{x} = f(x, t) = \begin{cases} \begin{pmatrix} \frac{1}{L} V_{AC} \sin \omega t - \frac{1}{L} v_{DC} \\ -\frac{1}{RC} v_{DC} + \frac{1}{C} i_{AC} \end{pmatrix}, & \text{falls } q(x, t) > 0, \\ \begin{pmatrix} \frac{1}{L} V_{AC} \sin \omega t + \frac{1}{L} v_{DC} \\ -\frac{1}{RC} v_{DC} - \frac{1}{C} i_{AC} \end{pmatrix}, & \text{falls } q(x, t) < 0. \end{cases}$$

Wenn zu einem Zeitpunkt  $t^*$  die Identität  $v_{\text{Soll}}(t^*) = v_{\text{DC}}(t^*)$  und die Abschätzung

$$|i_{\text{AC}}(t^*)| > \left| C\dot{v}_{\text{Soll}}(t^*) + \frac{1}{R}v_{\text{Soll}}(t^*) \right| \quad (3.2)$$

gilt, so lassen sich mit den Richtungsableitungen

$$\begin{aligned} D^+q(x, t) &= i_{\text{AC}} \left( \frac{1}{RC}v_{\text{DC}} - \frac{1}{C}i_{\text{AC}} + \dot{v}_{\text{Soll}}(t) \right), \\ D^-q(x, t) &= i_{\text{AC}} \left( \frac{1}{RC}v_{\text{DC}} + \frac{1}{C}i_{\text{AC}} + \dot{v}_{\text{Soll}}(t) \right) \end{aligned}$$

die Relationen  $D^+q(x^*, t^*) < 0$  und  $D^-q(x^*, t^*) > 0$  folgern. Um den Verlauf des Steuerungssignals in der anschließenden Lösungsphase zu bestimmen, betrachten wir allgemeine Systeme

$$\dot{x} = f_0(x, t) + b(x, t)u$$

mit glatten vektorwertigen Funktionen  $f_0, b : \mathbb{R}^{n_x+1} \rightarrow \mathbb{R}^{n_x}$ . Die skalare Steuerfunktion  $u$  sei unstetig:

$$u = \begin{cases} u^+, & \text{falls } q(x, t) > 0, \\ u^-, & \text{falls } q(x, t) < 0. \end{cases}$$

Im Vergleich zur allgemeinen unstetigen rechten Seite in Gleichung (3.1) schlägt sich hier die Unstetigkeit nur in der Steuerfunktion nieder. Die Berechnung einer Lösung in der Schaltfläche erfolgt mit dem „Equivalent Control“-Ansatz [UGS99, Chapter 2.3]:

$$0 = q_x f + q_t = q_x f_0 + q_x b u^{\text{eq}} + q_t.$$

Falls  $q_x b \neq 0$  gilt, dann folgt  $u^{\text{eq}} = -(q_x b)^{-1}(q_x f_0 + q_t)$  und damit

$$\dot{x} = f_0 - (q_x b)^{-1}(q_x f_0 + q_t)b.$$

Da die Steuerfunktion skalar ist und linear in der rechten Seite vorkommt, stimmt diese rechte Seite mit der für die Filippov-Lösung überein. Im Allgemeinen ist die „Equivalent Control“-Lösung nicht mit der Filippov-Lösung identisch.

Für den AC/DC-Konverter ergibt sich die Steuerfunktion gemäß

$$u^{\text{eq}} = \frac{RC\dot{v}_{\text{Soll}} + v_{\text{Soll}}}{Ri_{\text{AC}}},$$

woraus mit obiger Bedingung (3.2) die Abschätzung  $|u^{\text{eq}}| < 1$  folgt. Daher ist  $u^{\text{eq}}$  nur ein theoretisches Steuersignal, das garantieren würde, dass die Lösung in der Schaltfläche bliebe. In praktischen Anwendungen wird oft zugelassen, dass die Lösung in einer gewissen Umgebung der Schaltfläche bleibt. Im Gegenzug kommt man mit diskreten Werten  $u^+$  und  $u^-$  für  $u$  aus, die sich technisch realisieren lassen [AS98].  $\square$

## 3.2 Besonderheiten an Ereignispunkten

Die theoretischen Ergebnisse aus Abschnitt 3.1 helfen in praktischen Anwendungen, wenn sie eine genügend breite Klasse von Problemen abdecken. Es zeigt sich, dass in der Praxis zusätzliche Gegebenheiten mit berücksichtigt werden müssen. Die theoretischen Grundlagen dafür wollen wir in diesem Abschnitt legen. Zunächst lockern wir die Anforderungen an die Schaltfunktion, falls sie von Zustandsableitungen oder algebraischen Variablen abhängt und damit an Glattheitseigenschaften einbüßt. Im zweiten Teil schwächen wir die gängige Voraussetzung, dass zur Definition eines Ereignisses genau eine Schaltfunktion eine Nullstelle besitzt, ab.

### 3.2.1 Nichtglatte Schaltfunktionen

In der Darstellung von Definition 2.9 ist die Schaltfunktion  $q$  zustands- und zeitabhängig. In praktischen Modellen treten durchaus Schaltfunktionen  $\tilde{q}$  auf, die zusätzlich von den Zustandsableitungen  $\dot{x}$  oder algebraischen Variablen  $y$  abhängen:  $\tilde{q}(\dot{x}, y, x, t)$ . Eine solche Schaltfunktion kann mit der rechten Seite  $f$  des Differentialgleichungssystems und der Funktion  $g$  für die algebraischen Größen in  $q(x, t) := \tilde{q}(f(x, t), g(x, t), x, t)$  überführt werden. Allerdings ist dann nicht garantiert, dass  $q$  stetig bzw. noch glatter ist, da  $f$  in Definition 3.1 explizit abschnittsweise definiert ist und  $g$  durch die Abhängigkeit von Schaltern ebenfalls an Ereignispunkten nichtglatt sein kann. Dieser Abschnitt beschäftigt sich mit Lösungstechniken für Probleme mit nichtglatten Schaltfunktionen.

Die Gesamtheit aller Schaltfunktionen eines Modells sei im Vektor  $Q = (Q^1, \dots, Q^{n_Q})^\top$  zusammengefasst. Für jede Schaltfunktion  $Q^i$  verwenden wir einen Schalter  $S_i$ , so dass alle Schalter in  $S = (S_1, \dots, S_{n_Q})^\top \in \Omega^{n_Q}$  mit  $\Omega = \{-1, +1, 0\}$  verfügbar sind.

#### Definition 3.12 (Unstetige rechte Seite mit mehreren Schaltern)

Für  $S \in \Omega^{n_Q}$  sei eine glatte Abbildung  $f^S : \mathbb{R}^{n_x+1} \rightarrow \mathbb{R}^{n_x}$  gegeben. Die dem System

$$\dot{x} = f^S(x, t)$$

zugeordnete rechte Seite  $f^S$  bezeichnen wir dann als *unstetige rechte Seite mit mehreren Schaltern*. Die schalterabhängige Darstellung ist z. B. in [Eic92] zu finden. Analog lassen sich die algebraischen Größen mit  $g^S(x, t)$  beschreiben.  $\square$

#### Bemerkung 3.13 (Schalterabhängige Schaltfunktionen)

Da über  $\dot{x}$  und  $y$  Unstetigkeiten in den Schaltfunktionen induziert werden können, führen wir *schalterabhängige Schaltfunktionen* ein:

$$Q^{i,S}(x, t) := Q^i(f^S(x, t), g^S(x, t), x, t), \quad i = 1, \dots, n_Q.$$

Für  $Q^{i,S}(x(t), t) \neq 0$  gelte  $S_i = \text{sgn } Q^{i,S}(x(t), t)$ . Am Ereignispunkt  $(x^*, t^*)$  muss für die Schalter  $S$  entschieden werden, welche Werte sie für die Zeit  $t > t^*$  erhalten. Wir bezeichnen diese Schalterstellungen dann mit  $\bar{S} \in \Omega^{n_Q}$ . Sie übernehmen lokal für  $t > t^*$  die Rolle von  $S$ .

Tabelle 3.4: Resultierende Schalterkonstellationen  $\bar{S}$  für nichtglatte Schaltfunktionen. Die Schaltfunktion  $Q^1$  und die Richtungsableitungen  $D^\pm Q^1$  sind jeweils im Punkt  $(x^*, t^*)$  ausgewertet.

Fall	$\text{sgn } Q^{1,S^+}$	$\text{sgn } Q^{1,S^-}$	$\text{sgn } D^+ Q^{1,S^+}$	$\text{sgn } D^- Q^{1,S^-}$	$\bar{S}$
I	+1	+1	—	—	$S^+$
II	-1	-1	—	—	$S^-$
III	+1	0	—	+1	$S^+$
IV	0	-1	-1	—	$S^-$
V	0	0	+1	+1	$S^+$
VI	0	0	-1	-1	$S^-$
VII	0	0	-1	+1	$S^*$

Wenn für genau eine Komponente von  $Q^S$  (ohne Einschränkung für die erste) die Identität  $Q^{1,S}(x^*, t^*) = 0$  gilt, so folgt *nicht* notwendigerweise  $\bar{S}_i = S_i$  für  $i = 2, \dots, n_Q$ . Im Allgemeinen behalten nicht aktive Schaltfunktionen ihre Schalterstellung nur bei, wenn sie nicht von Schaltern abhängen.  $\square$

### Beispiel 3.14 (Verschachteltes Schalten)

Gegeben sei das folgende System:

$$z_1 = \begin{cases} 1, & \text{falls } x > 1, \\ -1, & \text{falls } x < 1, \end{cases} \quad z_2 = \begin{cases} 1, & \text{falls } z_1 > 0, \\ -1, & \text{falls } z_1 < 0, \end{cases} \quad \dot{x} = \begin{cases} 10, & \text{falls } z_2 > 0, \\ 100, & \text{falls } z_2 < 0 \end{cases}$$

mit  $x(0) = 0$  und den Schaltfunktionen  $Q^1 = x - 1$ ,  $Q^2 = z_1$ ,  $Q^3 = z_2$  sowie den zugehörigen Schaltern  $S_1 = S_2 = S_3 = -1$  für  $t = 0$ . Dann gilt:

$$x(t) = \begin{cases} 100t, & \text{falls } 0 \leq t < \frac{1}{100}, \\ 10(t - \frac{1}{100}) + 1, & \text{falls } \frac{1}{100} \leq t. \end{cases}$$

Am Ereigniszeitpunkt  $t^* = 1/100$  gilt  $Q^1 = 0$ , und die Schalter werden auf  $\bar{S}_1 = \bar{S}_2 = \bar{S}_3 = +1$  umgeschaltet. Es reicht also nicht aus, nur eine Schalterstellung am Ereignis zu ändern. Zu dem Beispiel ist das folgende System äquivalent, was sich allgemein eher schwer automatisch aus dem Originalsystem bestimmen lässt:

$$\dot{x} = \begin{cases} 10, & \text{falls } x > 1, \\ 100, & \text{falls } x < 1, \end{cases} \quad x(0) = 0.$$

$\square$

### Bemerkung 3.15 (Ereignis mit schalterabhängigen Schaltfunktionen)

Wir bestimmen nun die Schalterstellungen  $\bar{S}$  am Ereignispunkt. Es sei dazu  $S^+ \in \Omega^{n_Q}$  die Schalterkombination, die sich aus den Schaltern  $S$  und der Forderung  $S_1 = +1$  ergibt. Zur Bestimmung von  $S^+$  werden alle Schalter, die von  $S_1$  abhängen, neu berechnet.

Die restlichen bleiben unverändert. Analog erhält man  $S^-$  für  $S_1 = -1$ . Es wird angenommen, dass jeweils eine eindeutige Schalterkombination  $S^+$  und  $S^-$  existiert. Es gilt dann  $\text{sgn } Q^{i,S^\pm}(x^*, t^*) = S_i^\pm$  für  $i = 2, \dots, n_Q$ , wobei wir voraussetzen, dass  $S_i^\pm \neq 0$  erfüllt ist.

Wenn  $\text{sgn } Q^{1,S^+}(x^*, t^*) = 0$  oder  $\text{sgn } Q^{1,S^-}(x^*, t^*) = 0$  gelten, so können die Richtungsableitungen

$$D^+Q^{1,S^+} = Q_x^{1,S^+} f^{S^+} + Q_t^{1,S^+}, \quad D^-Q^{1,S^-} = Q_x^{1,S^-} f^{S^-} + Q_t^{1,S^-}$$

weitere Hinweise auf den konsistenten Schaltzustand  $\bar{S}$  lokal für  $t > t^*$  geben. Tabelle 3.4 listet die Fälle auf, in denen  $\bar{S}$  eindeutig bestimmt ist. Sie ergeben sich aus einfachen Überlegungen ähnlich denen in Abschnitt 3.1. Alle Fälle können als hybrides System geschrieben werden:

$$\dot{x}^0 = f^0(x^0, t) := f^S(x^0, t), \quad x^0(t^0) = A^0, \quad q^1(x^0, t) := Q^{1,S}(x^0, t),$$

$$x^1(t^1) = \text{id}_{x^0}, \quad \dot{x}^1 = f^1(x^1, t) := f^{\bar{S}}(x^1, t).$$

Allerdings lässt sich dieses System im Allgemeinen *nicht* wie in Bemerkung 3.8 auf ein System mit unstetiger rechter Seite und *glatter* Schaltfunktion zurückführen.

Für den Fall (VII) aus Tabelle 3.4 ist noch nicht klar, wie  $f^{\bar{S}}$  aussehen soll. Die Schalterkonstellation legt eine Filippov-Lösung nahe – jedoch sind die Voraussetzungen aus Definition 3.4 nicht erfüllt, da die Schaltfunktion  $Q^1$  nichtglatt sein kann. Trotzdem definieren wir die Schalterstellung  $S^* := (0, S_2, \dots, S_{n_q})^\top$ , die anzeigt, dass der Fall (VII) für die Schaltfunktion  $Q^1$  vorliegt. Diese Situation analysieren wir nun an einem Beispiel und dann allgemein genauer.  $\square$

### Beispiel 3.16 (Verlustbehaftetes Getriebe)

Wir untersuchen ein Modelica-Modell<sup>1</sup>, das ein mechanisches Übersetzungsgetriebe in einfacher Weise beschreibt. Die rotierenden Zahnräder werden mit mechanischen Trägheiten modelliert, die über ein ideales, verlustfreies Getriebe und einer auf den Wirkungsgrad bezogenen Drehmomentenlast miteinander gekoppelt sind, siehe Abbildung 3.4. Je nach Richtung des Leistungsflusses tritt der Wirkungsgrad  $\eta$  links bzw. rechts als Faktor in der Momentengleichung des Teilmodells **Wirkungsgrad** auf:

<pre> model Wirkungsgrad   extends TwoFlanges;   parameter Real eta = 0.5;   SIunits.Angle phi;   SIunits.Power power_a; equation   phi = flange_a.phi;   phi = flange_b.phi;   ... </pre>	<pre> ... power_a = flange_a.tau*der(phi); if power_a &gt;= 0 then   flange_b.tau = -eta*flange_a.tau; else   eta*flange_b.tau = -flange_a.tau; end if; end Wirkungsgrad; </pre>
--	--

<sup>1</sup>Der Autor ist C. Schweiger und M. Otter (DLR Oberpfaffenhofen) dankbar für die Bereitstellung des Modells.



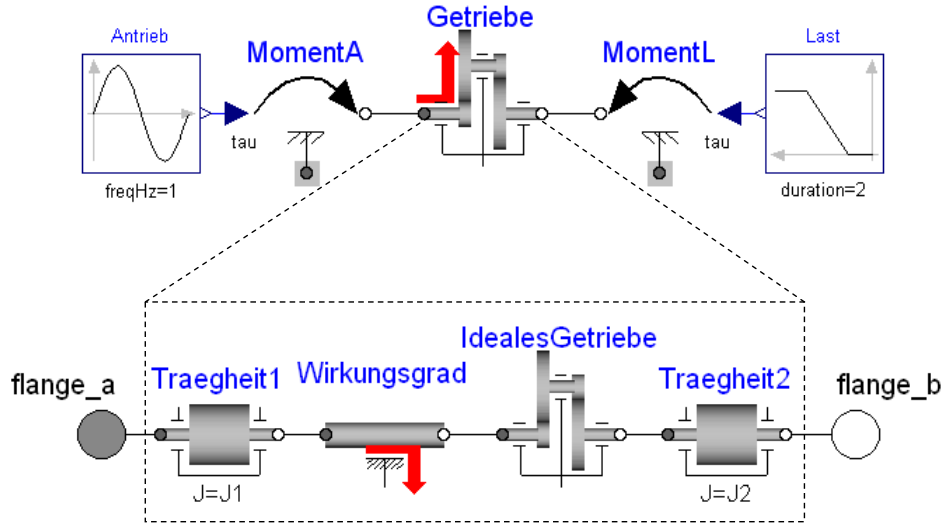


Abbildung 3.4: Modelica-Modell eines verlustbehafteten Getriebes mit Momentenaufschaltung.

Weitere Details zu dem Modell, etwa wie die Wirkungsgrade aus geometrischen Überlegungen bestimmt werden, findet man in [PSO02]. Dort werden auch Techniken eingeführt, die der entgegengesetzten Leistungsflussrichtung einen eigenen Wirkungsgrad zuordnen. Zudem können die Wirkungsgrade von der Drehzahl des Getriebes abhängen. Aus Gründen der Einfachheit verzichten wir auf diese Abhängigkeiten in unseren Ausführungen.

In Abbildung 3.4 ist ein Gesamtmodell zu sehen, mit dem sich die Eigenschaften des Getriebemodells analysieren lassen. Auf beiden mechanischen Wellen des Getriebes wird ein zeitabhängiges Drehmoment aufgebracht, um das Verhalten für verschiedenes Last- und Antriebsmoment zu prüfen. Alle wesentlichen Parameter und Variablen des Gesamtmodells sind mit ihren Werten in Tabelle 3.5 aufgelistet. Insgesamt erhält man folgendes System:

$$\dot{x} = \begin{cases} f^+(t), & \text{falls } q(x, t) > 0, \\ f^-(t), & \text{falls } q(x, t) < 0, \end{cases} \quad x(0) = 0$$

mit

$$f^+(t) := \frac{\eta r^2 M_A(t) + r M_L(t)}{\eta r^2 J_1 + J_2}, \quad f^-(t) := \frac{\frac{1}{\eta} r^2 M_A(t) + r M_L(t)}{\frac{1}{\eta} r^2 J_1 + J_2}$$

und

$$q(x, t) = x(M_A(t) - J_1 \dot{x}) = \begin{cases} x(M_A(t) - J_1 f^+(t)) =: q^+(x, t), & \text{falls } q > 0, \\ x(M_A(t) - J_1 f^-(t)) =: q^-(x, t), & \text{falls } q < 0. \end{cases}$$

Das Besondere an diesem Beispiel ist die Abhängigkeit der Schaltfunktion von  $\dot{x}$ , die dazu führt, dass sich die Schaltfunktion in zwei Äste aufspaltet:

$$q^+(x, t) = x \frac{J_2 M_A(t) - r J_1 M_L(t)}{\eta r^2 J_1 + J_2}, \quad q^-(x, t) = x \frac{J_2 M_A(t) - r J_1 M_L(t)}{\frac{1}{\eta} r^2 J_1 + J_2}.$$

Tabelle 3.5: Parameter und Variablen für Beispiel 3.16, Abbildung 3.4.

Bez.	Modelica-Variable	Beschreibung	Wert	Einheit
$J_1$	Getriebe.J1	Trägheitsmoment, linkes Zahnrad	1	kg m <sup>2</sup>
$J_2$	Getriebe.J2	Trägheitsmoment, rechtes Zahnrad	1.5	kg m <sup>2</sup>
$r$	Getriebe.r	Übersetzungsverhältnis	2	1
$\eta$	Getriebe.eta	Wirkungsgrad	0.5	1
$M_A$	MomentA.tau	Drehmoment, links	$10 \sin(2\pi t)$	Nm
$M_L$	MomentL.tau	Drehmoment, rechts	$-10 + 2.5t$	Nm
$\omega_1$	Getriebe.w1	Drehgeschwindigkeit, links	$x(t)$	rad/s
$\omega_2$	Getriebe.w2	Drehgeschwindigkeit, rechts	$\omega_2(t)$	rad/s

Wir stellen fest, dass stets  $\operatorname{sgn} q^+ = \operatorname{sgn} q^-$  gilt, weil die Nenner in den Ausdrücken für  $q^+$  und  $q^-$  jeweils positiv sind. Insbesondere ist  $q^+ = 0$  und  $q^- = 0$  auf derselben Menge und damit  $\Sigma$  wohldefiniert. Somit gibt es stets einen eindeutigen Wert für  $q(x, t)$ . Abbildung 3.5 zeigt den Verlauf von  $q$ , wobei die Bereiche für  $q > 0$ ,  $q < 0$  und  $q = 0$  farblich unterschiedlich gekennzeichnet sind. Wegen  $\operatorname{sgn} q^+ = \operatorname{sgn} q^-$  ist  $q$  stetig, aber nicht notwendigerweise differenzierbar. Betrachtet man die Ableitungen

$$\begin{aligned}
q_x^+(x, t) &= \frac{J_2 M_A(t) - r J_1 M_L(t)}{\eta r^2 J_1 + J_2}, & q_x^-(x, t) &= \frac{J_2 M_A(t) - r J_1 M_L(t)}{\frac{1}{\eta} r^2 J_1 + J_2}, \\
q_t^+(x, t) &= x \frac{J_2 \dot{M}_A(t) - r J_1 \dot{M}_L(t)}{\eta r^2 J_1 + J_2}, & q_t^-(x, t) &= x \frac{J_2 \dot{M}_A(t) - r J_1 \dot{M}_L(t)}{\frac{1}{\eta} r^2 J_1 + J_2},
\end{aligned}$$

so wird klar, dass im Allgemeinen für  $\eta \neq 1$  die Aussagen  $q_x^+ \neq q_x^-$  und  $q_t^+ \neq q_t^-$  gelten.  $\square$

Wir lösen uns jetzt von Beispiel 3.16 und betrachten allgemein unstetige Systeme mit regulärer Schaltfunktion. In Beispiel 3.22 kommen wir noch einmal auf das Modell des verlustbehafteten Getriebes zurück.

### Bemerkung 3.17 (Darstellung als System mit unstetiger rechter Seite)

Zu den Gegebenheiten aus Bemerkung 3.15 seien für den Fall (VII) aus Tabelle 3.4 die Definitionen  $q^+ := Q^{1, S^+}$ ,  $q^- := Q^{1, S^-}$ ,  $f^+ := f^{S^+}$  und  $f^- := f^{S^-}$  vereinbart. Da wir nur an Situationen interessiert sind, die eine eindeutige Lösung ermöglichen, ist es sinnvoll, an die Äste der aktiven Schaltfunktion die Voraussetzung  $\operatorname{sgn} q^+ = \operatorname{sgn} q^-$  in  $\mathbb{R}^{n_x+1}$  zu stellen. Damit sind die Schaltfunktion

$$q(x, t) := \begin{cases} q^+(x, t), & \text{falls } q^+(x, t) > 0, \\ q^-(x, t), & \text{falls } q^-(x, t) < 0 \end{cases}$$

und die Schaltfläche  $\Sigma$  wohldefiniert, insbesondere ist  $q$  stetig. Es liegt also ein System mit unstetiger rechter Seite nach Definition 3.1 und mit einer regulären Schaltfunktion nach folgender Definition vor.  $\square$

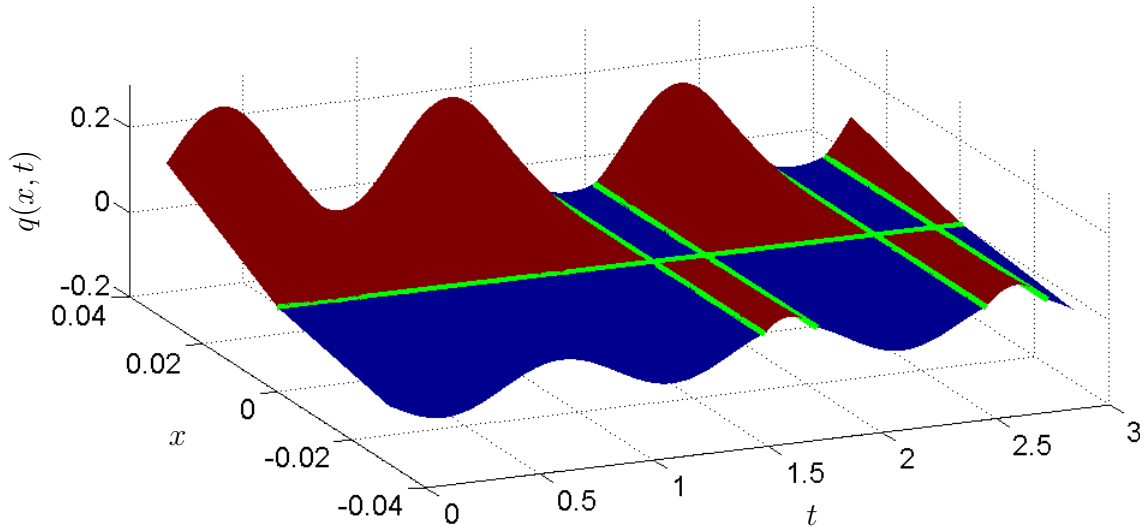


Abbildung 3.5: Nichtglatter Schaltfunktionsverlauf  $q(x, t)$  für Beispiel 3.16. Grüne Kurven (Punkte, an denen  $q$  nicht differenzierbar ist):  $q = q^+ = q^- = 0$ . Braune Fläche:  $q = q^+$ . Blaue Fläche:  $q = q^-$ .

### Definition 3.18 (Reguläre Schaltfunktion)

Es sei  $q$  eine stetige Schaltfunktion, die außerhalb von  $\Sigma$  glatt sei und folgende Eigenschaft erfülle: Es existiert eine Funktion  $q^+$ , für die  $q^+ = q$  auf  $\Sigma^+ := \{(x, t) : q(x, t) > 0\}$  gilt und die auf  $\Sigma^+ \cup U(\Sigma)$  glatt ist.  $U(\Sigma)$  ist hierbei eine hinreichend kleine Umgebung von  $\Sigma$ . Ebenso existiert eine Funktion  $q^-$  mit analogen Eigenschaften auf  $\Sigma^- \cup U(\Sigma)$ . Wir bezeichnen die Schaltfunktion dann als *reguläre Schaltfunktion*.  $\square$

### Lemma 3.19 (Lineare Abhängigkeit der Normalenvektoren von $\Sigma$ )

Es sei eine reguläre Schaltfunktion  $q$  gegeben. Dann gibt es zu jedem Punkt  $(x, t) \in \Sigma$  mit  $\nabla q^+(x, t) \neq 0$ ,  $\nabla q^-(x, t) \neq 0$  eine positive Zahl  $\kappa > 0$ , für die folgendes gilt:

$$\nabla q^+(x, t) = \kappa \nabla q^-(x, t).$$

### Beweis:

Unter den gegebenen Voraussetzungen spannt sowohl  $\nabla q^+(x, t)$  als auch  $\nabla q^-(x, t)$  den Normalenraum zur Schaltfläche  $\Sigma$  im Punkt  $(x, t)$  auf [Kön97, Kapitel 3.5]. Da der Normalenraum ein eindimensionaler Unterraum von  $\mathbb{R}^{n_x+1}$  ist, folgt, dass ein  $\kappa \neq 0 \in \mathbb{R}$  existiert, mit dem die Gleichung aus dem Lemma gilt. Da die Gradienten nicht verschwinden, sind  $\Sigma^+ \neq \emptyset$  und  $\Sigma^- \neq \emptyset$ . Wegen der allgemeinen Eigenschaften eines Gradienten [Kön97, Kapitel 2.1] zeigt  $\nabla q^+$  im Punkt  $(x, t)$  in die Menge  $\Sigma^+$  hinein, ebenso zeigt  $-\nabla q^-$  in die Menge  $\Sigma^-$ , siehe auch Abbildung 3.6. Somit ergibt sich  $\kappa > 0$ .  $\blacksquare$

### Korollar 3.20 (Vorzeichen der Richtungsableitungen)

Für  $\nabla q^+(x, t) \neq 0$ ,  $\nabla q^-(x, t) \neq 0$  gelten mit den Richtungsableitungen lokal um  $(x, t) \in \Sigma$ :

$$\operatorname{sgn} D^+ q^+ = \operatorname{sgn} D^+ q^- \quad \text{sowie} \quad \operatorname{sgn} D^- q^- = \operatorname{sgn} D^- q^+.$$

$\blacksquare$

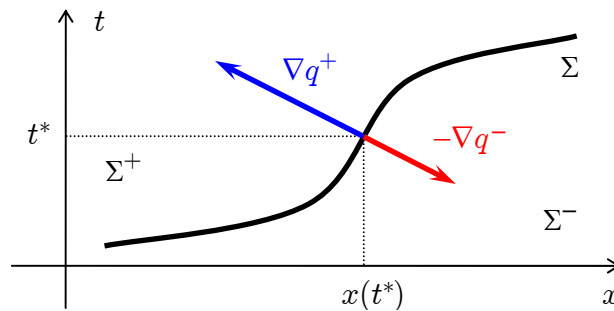


Abbildung 3.6: Gradienten von  $q^+$ ,  $q^-$  an der Schaltfläche  $\Sigma$  für eine reguläre Schaltfunktion  $q$ .

### Satz 3.21 (Filippov-Lösung für reguläre Schaltfunktionen)

Es sei ein System mit unstetiger rechter Seite und regulärer Schaltfunktion gegeben. Gelten für die Richtungsableitungen die Relationen  $D^+q^+(x^*, t^*) < 0$  und  $D^-q^-(x^*, t^*) > 0$ , dann existiert für  $t > t^*$  eine eindeutige Lösung mit der Ableitung

$$\dot{x} = f^*(x, t) := \frac{D^+q^+ f^- - D^-q^+ f^+}{D^+q^+ - D^-q^+}(x, t) = \frac{D^+q^- f^- - D^-q^- f^+}{D^+q^- - D^-q^-}(x, t).$$

Sie bleibt solange gültig, bis  $D^+q^+$  oder  $D^-q^-$  verschwinden, falls  $x$  beschränkt bleibt. Wir nennen die Lösung ebenfalls Filippov-Lösung.

#### Beweis:

Durch die Vorzeichensituation von  $D^+q^+$  und  $D^-q^-$  ist klar, dass keine Lösung mit  $q > 0$  oder  $q < 0$  für  $t > t^*$  existiert. Zudem folgt  $\nabla q^+(x^*, t^*) \neq 0$ ,  $\nabla q^-(x^*, t^*) \neq 0$ . Mit Lemma 3.19 ergibt sich  $D^+q^+(x^*, t^*) = \kappa D^+q^-(x^*, t^*)$  und  $D^-q^+(x^*, t^*) = \kappa D^-q^-(x^*, t^*)$  für ein  $\kappa > 0$ . Damit ist  $f^*$  wohldefiniert, da die Definition von  $f^*$  unabhängig davon ist, ob  $q^+$  oder  $q^-$  als definierende Schaltfunktion verwendet wird. Daher gibt es die so definierte Filippov-Lösung, falls sie beschränkt bleibt, zeitlich mindestens, bis  $D^+q^+$  oder  $D^-q^+$  verschwinden. Nach Korollar 3.20 ist das äquivalent dazu, dass  $D^+q^-$  oder  $D^-q^-$  zu Null werden. ■

### Beispiel 3.22 (Verlustbehaftetes Getriebe – Fortsetzung von Beispiel 3.16)

Beispiel 3.16 stellt ein System mit unstetiger rechter Seite und regulärer Schaltfunktion im obigen Sinn dar. Deswegen wird die Analyse dieses Beispiels gemäß den allgemeinen Überlegungen fortgeführt.

Zur Zeit  $t = 0$  gilt  $q^+(0, 0) = q^-(0, 0) = 0$  sowie

$$q_x^+(0, 0) = \frac{40}{7}, \quad q_t^+(0, 0) = 0, \quad f^+(0) = -\frac{40}{7}, \quad f^-(0) = -\frac{40}{19}.$$

Damit folgt:  $D^+q^+(0, 0) = -1600/49 < 0$  und  $D^-q^+(0, 0) = -1600/133 < 0$ . Also wird die Lösung lokal für  $t > 0$  mit

$$f^-(t) = \frac{160 \sin 2\pi t - 40 + 10t}{19}$$

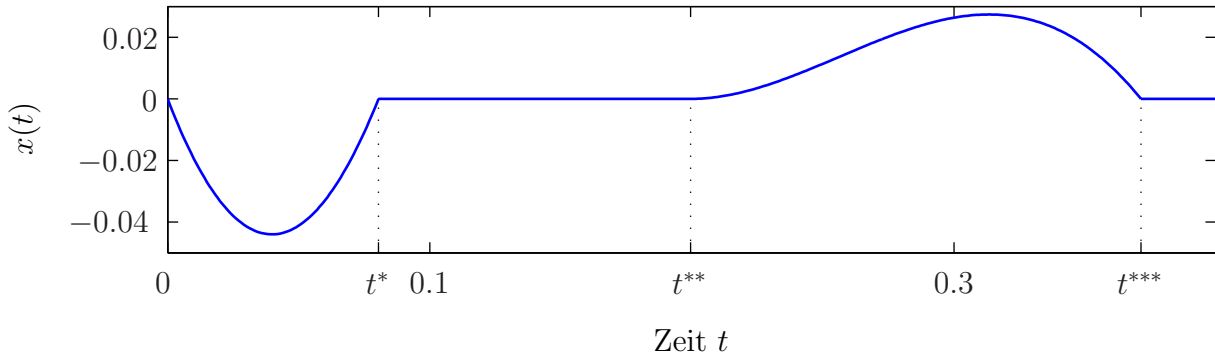


Abbildung 3.7: Berechnete Lösung  $x(t)$  von Beispiel 3.16 und 3.22 mit regulärer Schaltfunktion. Die konstanten Phasen resultieren aus Filippov-Lösungen.

fortgesetzt. Die zugehörige Schaltfunktion

$$q^-(x, t) = x \frac{30 \sin 2\pi t + 40 - 10t}{19}$$

wird für  $t \in (0, 1.5)$  genau dann zu Null, wenn auch  $x$  verschwindet. Die Lösung

$$x(t) = \frac{-\frac{80}{\pi}(\cos 2\pi t - 1) - 40t + 5t^2}{19}$$

hat die erste Nullstelle  $t^* > 0$  für  $t^* = 0.080477$ . Damit gilt  $q^-(0, t^*) = 0$ ,  $q_x^-(0, t^*) = 2.8277$  und  $q_t^-(0, t^*) = 0$ . Die Richtungsableitungen haben die Werte  $D^+q^-(0, t^*) = -8.0065 < 0$  und  $D^-q^-(0, t^*) = 5.70087 > 0$ . Aus Satz 3.21 folgt, dass eine Filippov-Lösung existiert, wobei  $f^* = 0$  und damit  $x(t) = 0$  lokal für  $t > t^*$  gelten. Der erste Nulldurchgang der neuen Schaltfunktionen  $D^+q^-$  und  $D^-q^-$  tritt bei  $t^{**} = 0.1995$  für  $D^+q^-$  auf. Hier verlässt die Lösung die Schaltfläche mit  $f = f^+$  lokal für  $t > t^{**}$ :

$$x(t) = \frac{\frac{-20}{\pi}(\cos 2\pi t - \cos 2\pi t^{**}) - 40(t - t^{**}) + 5(t^2 - t^{**2})}{7}.$$

Die nächste Nullstelle von  $q^+$  ist bei  $t^{***} = 0.3713$ . Ab hier bleibt die Lösung dann wieder für eine bestimmte Zeit in der Schaltfläche. In Abbildung 3.7 ist der Lösungsverlauf dargestellt, den weiteren Verlauf sehen wir in Abbildung 5.3 aus Kapitel 5 als Ergebnis einer numerischen Simulation.

Die errechnete Lösung lässt sich physikalisch wie folgt interpretieren: Obwohl sich die auf das Getriebe aufgebrachten Momente kontinuierlich ändern, bewegen sich die Anschlussflansche des Getriebes im Zeitintervall  $[t^*, t^{**}]$  nicht. Verluste im Getriebe hemmen eine Bewegung in beide Drehrichtungen solange, bis das Moment an einer Getriebeseite betragsmäßig groß genug ist, damit es als Antriebsmoment dienen und das Abtriebsmoment inklusive Verluste aufbringen kann. Dieses Verhalten ist dem eines Quaders auf einer ebenen Fläche recht ähnlich, vgl. Beispiel 3.10. Eine Bewegung des Quaders setzt erst dann ein, wenn die maximale Haftreibungskraft durch die Anregungskraft überwunden wird.  $\square$

### 3.2.2 Mehrere aktive Schaltfunktionen bei rekursiver Staffelung

In Abschnitt 3.1 wurde stets vorausgesetzt, dass genau eine skalare Schaltfunktion  $q$  zur Zeit  $t^*$  einen Nulldurchgang hat, was eine übliche Annahme [ST00] ist. Wir wollen nun Systeme untersuchen, bei denen mehrere Schaltfunktionen zur gleichen Zeit Nullstellen besitzen. In [Fri04, Chapter 13.2.6.6] wird unterschieden zwischen Ereignissen, die direkt durch Nullstellen mehrerer Schaltfunktionen zur selben Zeit erzeugt werden und solchen, die als Folge eines momentanen Ereignisses durch interne Modellschaltungen zustande kommen. Der zweite Fall wurde bereits teilweise mit den schalterabhängigen Schaltfunktionen im vorangegangenen Abschnitt behandelt. Daher beschäftigen wir uns zunächst nur mit der ersten Variante und setzen in diesem Abschnitt voraus, dass die Schaltfunktionen hinreichend glatt sind. Bemerkung 3.29 erweitert schließlich die Ergebnisse des Abschnitts auf nichtglatte reguläre Schaltfunktionen. Wir betrachten in dem unstetigen System mit mehreren Schaltern (vgl. Definition 3.12)

$$\dot{x} = f^s(x, t) \quad (3.3)$$

nur die Abhängigkeit von den Schaltern  $s \in \Omega^{n_q}$  der *aktiven* Schaltfunktionen  $q = (q^1, \dots, q^{n_q})^\top : \mathbb{R}^{n_x+1} \rightarrow \mathbb{R}^{n_q}$ . Die gleichzeitig aktiven Schaltfunktionen in den Komponenten von  $q$  erfordern eine detailliertere Untersuchung der Struktur des Gesamtsystems. An einem Ereignispunkt besteht das Problem, eine neue glatte rechte Seite  $\phi : \mathbb{R}^{n_x+1} \rightarrow \mathbb{R}^{n_x}$  zu definieren, die lokal für  $t > t^*$  die Dynamik des Systems korrekt beschreibt. Dafür müssen die Schalter  $s$  für die Zeit  $t > t^*$  neue Werte  $\bar{s} \in \Omega^{n_q}$  erhalten. Wir zeigen, wie die Analyse für Systeme möglich ist, die wie folgt rekursiv auflösbar sind.

#### Bemerkung 3.23 (Rekursiv gestaffelte Systeme)

Es sei  $q(x^*, t^*) = 0$ . Wir nehmen an, dass sich das System (3.3) nach einer evtl. Umnummerierung der Komponenten von  $x$  und  $q$  in der folgenden Form darstellen lässt:

$$x = \begin{pmatrix} \xi^0 \\ \xi^1 \\ \vdots \\ \xi^m \end{pmatrix}, \quad f^s(x, t) = \begin{pmatrix} \varphi^0(x, t) \\ \varphi^{1,s^1}(x, t) \\ \vdots \\ \varphi^{m,s^m}(x, t) \end{pmatrix}, \quad s^k = \begin{pmatrix} s_1^0 \\ \vdots \\ s_\ell^0 \\ s_1 \\ \vdots \\ s_k \end{pmatrix}$$

mit  $s_i^0 := s_{m+i}$  für  $i = 1, \dots, \ell$ ,

$$\begin{aligned} q^k &= q^k(\xi^0, \dots, \xi^k, t) && \text{für } k = 1, \dots, m, \\ q^i &= q^i(\xi^0, t) && \text{für } i = m+1, \dots, m+\ell, \\ q^i &= q^i(x, t) && \text{für } i = m+\ell+1, \dots, n_q. \end{aligned}$$

Es gelte dabei  $m + \ell \leq n_q$  für ganze Zahlen  $m, \ell \geq 0$ . Wir lassen  $n_{\xi^0} = 0$  zu, für  $k > 0$  gelte  $n_{\xi^k} > 0$ . Wir erhalten also das Teilsystem  $\dot{\xi}^0 = \varphi^0(x, t)$ , das den glatten Anteil des Gesamtsystems an diesem Ereignispunkt enthält, und die Teilsysteme

$$\dot{\xi}^k = \varphi^{k,s^k}(x, t) \quad \text{für } k = 1, \dots, m. \quad (3.4)$$

Die aktiven Schaltfunktionen unterteilen sich in drei Gruppen. Die erste Gruppe von Schaltfunktionen  $q^k$  ( $k = 1, \dots, m$ ) lässt die Abhängigkeit von  $\xi^0, \dots, \xi^k$  sukzessive zu. Die Schaltfunktionen  $q^i$  ( $i = m + 1, \dots, m + \ell$ ) der zweiten Gruppe hängen nur von  $\xi^0$  und  $t$  ab. Die zugehörigen Schalter können auf alle rechten Seiten  $\varphi^{k,s^k}$  Einfluss nehmen. In diese Gruppe fallen insbesondere Schaltfunktionen, die nur von der Zeit abhängen. Für die dritte Klasse von Schaltfunktionen  $q^i$  ( $i = m + \ell + 1, \dots, n_q$ ) sind keine Einschränkungen in den Argumenten vorgesehen. Die rechte Seite hängt allerdings auch nicht von den Schaltern der letzten Gruppe ab. Diese Schalter beeinflussen beispielsweise algebraische Größen. Aus den Forderungen an  $q$  folgt für  $k = 1, \dots, m - 1$  die wichtige Feststellung:

$$q_{\xi^r}^k = 0 \quad \text{mit } k < r \leq m. \quad (3.5)$$

Der Aufteilung von  $x$  entsprechend unterteilen wir die neue, zu bestimmende rechte Seite in

$$\phi(x, t) = (\phi^0(x, t)^\top, \phi^1(x, t)^\top, \dots, \phi^m(x, t)^\top)^\top.$$

□

### Bemerkung 3.24 (Neue Schalterstellungen und neue rechte Seite)

Wir berechnen für rekursiv gestaffelte Systeme gemäß Bemerkung 3.23 die rechte Seite  $\phi$  und die Schalterstellungen  $\bar{s}$ . Es sei dazu  $\phi^0 := \varphi^0$ . Die Schalter  $\bar{s}_i^0$  ( $i = 1, \dots, \ell$ , falls  $\ell > 0$ ) können über

$$D^+q^{m+i}(x^*, t^*) = D^-q^{m+i}(x^*, t^*) = [q_{\xi^0}^{m+i}\phi^0 + q_t^{m+i}] (x^*, t^*)$$

und  $\bar{s}_i^0 := \text{sgn } D^+q^{m+i}(x^*, t^*)$  entkoppelt voneinander bestimmt werden, vgl. Tabelle 3.1. Es sei dabei vorausgesetzt, dass  $D^+q^{m+i}(x^*, t^*) \neq 0$  gilt.

Für  $k = 1, \dots, m$  berechnen wir dann wegen Gleichung (3.5) rekursiv die Richtungsableitungen

$$D^\pm q^k(x^*, t^*) = \left[ \sum_{r=0}^{k-1} q_{\xi^r}^k \phi^r + q_{\xi^k}^k \varphi^{k, \bar{s}_\pm^k} + q_t^k \right] (x^*, t^*)$$

mit  $\bar{s}_\pm^k := (\bar{s}_1^0, \dots, \bar{s}_\ell^0, \bar{s}_1, \dots, \bar{s}_{k-1}, \pm 1)^\top$ . Über die Vorzeichen der Richtungsableitungen wird der Wert für  $\bar{s}_k$  definiert. Damit ergibt sich für  $\bar{s}_k = \pm 1$  die Definition

$$\phi^k := \varphi^{k, \bar{s}_\pm^k} \quad \text{und für } \bar{s}_k = 0 : \quad \phi^k := \frac{D^+q^k \varphi^{k, \bar{s}_-^k} - D^-q^k \varphi^{k, \bar{s}_+^k}}{D^+q^k - D^-q^k}.$$

Damit ist  $\phi$  vollständig definiert. Als neue Schaltfunktionen dienen zusätzlich  $D^+q^k$  und  $D^-q^k$ , falls  $\bar{s}_k = 0$  gilt (Filippov-Lösung). Sie ersetzen die Schaltfunktion  $q^k$ . Die restlichen Schalter  $\bar{s}_i$  ( $i = m + \ell + 1, \dots, n_q$ , falls  $m + \ell < n_q$ ) können über

$$D^+q^i(x^*, t^*) = D^-q^i(x^*, t^*) = [q_x^i \phi + q_t^i] (x^*, t^*)$$

und  $\bar{s}_i := \text{sgn } D^+q^i(x^*, t^*)$  wiederum entkoppelt voneinander berechnet werden. Hierfür ist allerdings die Kenntnis der neuen rechte Seite  $\phi$  erforderlich. Es gelte  $D^+q^i(x^*, t^*) \neq 0$ . Für die Schaltfunktionen  $q^i$ ,  $i = m + 1, \dots, n_q$  können die Richtungsableitungen keine verschiedenen Vorzeichen annehmen, weil stets  $D^+q^i = D^-q^i$  gilt.

In der gezeigten Lösungskonstruktion ist es denkbar, dass Filippov-Lösungen für verschiedene Teilsysteme  $k$  zeitlich parallel auftreten. In den numerischen Untersuchungen in den Kapiteln 5 und 6 sind derartige Beispiele aufgeführt. □

**Beispiel 3.25 (Gestaffeltes System mit  $m = 2$ ,  $\ell = 1$ ,  $n_q = 3$ )**

Zur Verdeutlichung der Nomenklatur aus Bemerkung 3.24 notieren wir eine beispielhafte Situation am Ereignis für drei aktive Schaltfunktionen. Es seien

$$x = \begin{pmatrix} \xi^0 \\ \xi^1 \\ \xi^2 \end{pmatrix} \text{ mit } f^s(x, t) = \begin{pmatrix} \varphi^0(x, t) \\ \varphi^{1, (s_1^0, s_1)}(x, t) \\ \varphi^{2, (s_1^0, s_1, s_2)}(x, t) \end{pmatrix}$$

sowie  $q^1 = q^1(\xi^0, \xi^1, t)$ ,  $q^2 = q^2(x, t)$  und  $q^3 = q^3(\xi^0, t)$  gegeben. Es ergibt sich für den ersten Schalter  $\bar{s}_1^0 := \text{sgn}[q_{\xi^0}^3 \phi^0 + q_t^3](x^*, t^*)$ . Dann berechnen sich die Richtungsableitungen für  $q^1$  zu

$$D^\pm q^1(x^*, t^*) = \left[ q_{\xi^0}^1 \varphi^0 + q_{\xi^1}^1 \varphi^{1, (\bar{s}_1^0, \pm 1)} + q_t^1 \right] (x^*, t^*),$$

da  $\varphi^1$  nicht von  $s_2$  und  $q^1$  nicht von  $\xi^2$  abhängen. Aus den Vorzeichen dieser Richtungsableitungen erhält man  $\bar{s}_1$ . Weiter ist

$$D^\pm q^2(x^*, t^*) = \left[ q_{\xi^0}^2 \varphi^0 + q_{\xi^1}^2 \varphi^{1, (\bar{s}_1^0, \bar{s}_1)} + q_{\xi^2}^2 \varphi^{2, (\bar{s}_1^0, \bar{s}_1, \pm 1)} + q_t^2 \right] (x^*, t^*)$$

und damit  $\bar{s}_2$  zu berechnen, da  $\bar{s}_1^0, \bar{s}_1$  bereits bekannt sind. □

**Bemerkung 3.26 (Reduziert gekoppelte Systeme)**

Als Spezialfall der obigen Staffelung betrachten wir ein System, das sich wie folgt darstellen lässt:

$$f^s(x, t) = \begin{pmatrix} \varphi^0(x, t) \\ \varphi^{1, \sigma^1}(x, t) \\ \vdots \\ \varphi^{m, \sigma^m}(x, t) \end{pmatrix}, \quad \sigma^k = \begin{pmatrix} s_1^0 \\ \vdots \\ s_\ell^0 \\ s_k \end{pmatrix}, \quad \begin{array}{ll} q^k = q^k(\xi^0, \xi^k, t) & \text{für } k = 1, \dots, m, \\ q^i = q^i(\xi^0, t) & \text{für } i = m + 1, \dots, m + \ell, \\ q^i = q^i(x, t) & \text{für } i = m + \ell + 1, \dots, n_q. \end{array}$$

Der Unterschied zum allgemeinen rekursiv gestaffelten System besteht darin, dass die Bestandteile  $\varphi^{k, \sigma^k}$  der rechten Seite neben den Schaltern  $s_i^0$  ( $i = 1, \dots, \ell$ ) nur vom Schalter *einer* zusätzlichen aktiven Schaltfunktion abhängen. Des Weiteren darf diese Schaltfunktion nur von den Zuständen  $\xi^0$  und  $\xi^k$  abhängig sein. Durch diese reduzierte Kopplung ergibt sich für die Berechnung der Richtungsableitungen die einfachere Gleichung:

$$D^\pm q^k(x^*, t^*) = \left[ q_{\xi^0}^k \varphi^0 + q_{\xi^k}^k \varphi^{k, \bar{\sigma}^\pm} + q_t^k \right] (x^*, t^*).$$

Die Schalterstellungen lassen sich in den drei aufeinander aufbauenden Schritten  $\bar{s}_i^0$  ( $i = 1, \dots, \ell$ ),  $\bar{s}_k$  ( $k = 1, \dots, m$ ) und schließlich  $\bar{s}_i$  ( $i = m + \ell + 1, \dots, n_q$ ) berechnen. Innerhalb der Schritte können die Schalterstellungen jeweils unabhängig voneinander ermittelt werden, da aus den Teilsystemen  $\xi^k = \varphi^{k, \sigma^k}(x, t)$  die Werte für  $\bar{s}_k$  und damit  $\phi^k$  parallel bestimmbar sind. □

**Korollar 3.27 (Einfach gekoppelte Systeme)**

Wir betrachten die noch weiter vereinfachte Struktur

$$f^s(x, t) = \begin{pmatrix} \varphi^0(x, t) \\ \varphi^{1, s_1}(x, t) \\ \vdots \\ \varphi^{m, s_m}(x, t) \end{pmatrix}, \quad \begin{array}{ll} q^k = q^k(\xi^0, \xi^k, t) & \text{für } k = 1, \dots, m, \\ q^i = q^i(\xi^0, t) & \text{für } i = m + 1, \dots, n_q. \end{array}$$



Bei diesem System hängt jede Komponente der rechten Seite höchstens von einem Schalter ab, der einer aktiven Schaltfunktion zugeordnet ist. Die Schaltfunktionen  $q^i$ , deren Schalter nicht auf die rechte Seite wirken, seien nur von  $\xi^0$  und  $t$  abhängig. Dann gilt für  $k = 1, \dots, m$ :

$$D^\pm q^k(x^*, t^*) = [q_{\xi^0}^k \varphi^0 + q_{\xi^k}^k \varphi^{k, \pm 1} + q_t^k](x^*, t^*) = [q_x^k f^{s(k \pm)} + q_t^k](x^*, t^*)$$

mit  $s(k \pm) = (s_1, \dots, s_{k-1}, \pm 1, s_{k+1}, \dots, s_{n_q})$ . Folglich können alle Schalterstellungen  $\bar{s}_i$  ( $i = 1, \dots, n_q$ ) unabhängig voneinander berechnet werden. Diese Struktur wird dann in Kapitel 5.2 Grundlage für die Implementierung der numerischen Lösungsverfahren sein. ■

### Beispiel 3.28 (Belasteter Elektromotor – unstetiges Lastmoment)

In Beispiel 3.9 wird das Lastdrehmoment  $M_L$  des Elektromotors als glatte Funktion modelliert. Nun schwächen wir die Voraussetzung ab und betrachten einen sprunghaften Verlauf:

$$M_L(t) := \begin{cases} 0, & \text{falls } t < \tau, \\ \bar{M}, & \text{falls } \tau < t. \end{cases}$$

Es sei dabei  $\bar{M} = -40 \text{ Nm}$  und  $\tau > 0$  der Zeitpunkt, ab dem das Moment  $\bar{M}$  auf den Motor wirkt. Gilt nun  $v_G(t) = v_C(t)$  genau zur Zeit  $t = \tau$ , dann liegt ein einfach gekoppeltes System wie in Korollar 3.27 mit  $m = n_q = 2$  vor:  $\xi^0 := i_L$ ,  $\xi^1 := v_C$ ,  $\xi^2 := w$ ,  $q^1(\xi^1, t) := v_G(t) - v_C = v_G(t) - \xi^1$ ,  $q^2(t) = t - \tau$ ,

$$\begin{pmatrix} \dot{\xi}^0 \\ \dot{\xi}^1 \\ \dot{\xi}^2 \end{pmatrix} = \begin{pmatrix} \varphi^0(x, t) \\ \varphi^{1, s_1}(x, t) \\ \varphi^{2, s_2}(x, t) \end{pmatrix} = \begin{pmatrix} -\frac{R}{L}\xi^0 + \frac{1}{L}\xi^1 - \frac{k}{L}\xi^2 \\ -\frac{1}{C}\xi^0 + a_{11}^{s_1}\xi^1 + b_1^{s_1}(t) \\ \frac{k}{J}\xi^0 + \frac{1}{J}M_L^{s_2}(t) \end{pmatrix}.$$

Für die Bezeichnungen sei auf Beispiel 3.9 mit Tabelle 3.3 auf Seite 26 verwiesen. □

### Bemerkung 3.29 (Mehrere aktive reguläre Schaltfunktionen)

Im Abschnitt 3.2.2 wurde bisher vorausgesetzt, dass die aktiven Schaltfunktionen glatt seien. Mit Hilfe von Korollar 3.20 und Satz 3.21 können die Aussagen auf Systeme mit den beschriebenen Strukturen – aber mit regulären Schaltfunktionen – übertragen werden. Für jede aktive reguläre Schaltfunktion  $q$  muss man sich lediglich auf einen Ast  $q^+$  oder  $q^-$  festlegen und diesen konsequent für alle Berechnungen, in denen  $q = 0$  gilt, verwenden. Aus Gründen der Lesbarkeit soll die Erweiterung auf mehrere aktive (möglicherweise nichtglatte) reguläre Schaltfunktionen hier jedoch nicht im Detail diskutiert werden. □

Kapitel 3 gibt einen tiefen Einblick, welche Situationen an Ereignispunkten hybrider dynamischer Systeme zustande kommen können. Neben dem klassischen Umschalten einer unstetigen rechten Seite führen inkonsistente Schaltzustände zu Filippov-Lösungen, die auch für eine Unterklasse nichtglatter Schaltfunktionen existieren. Rekursiv gestaffelte Systeme und entsprechende Spezialfälle lassen eindeutige Lösungen mit mehreren aktiven Schaltfunktionen zu.

## 4 Sensitivitätsanalyse unstetiger Systeme

Neben der Lösung eines dynamischen Systems interessiert häufig auch die Empfindlichkeit der Lösung gegenüber kleinen Änderungen von Modellparametern  $p \in \mathbb{R}^{n_p}$ . Praktische Modelle komplexer Systeme enthalten meistens zahlreiche Parameter, die eine Feinjustierung des Modells ermöglichen. Aus analytischer Sicht beschreiben Ableitungen der Lösung nach den Parametern diese Empfindlichkeiten, die lokal in einer Umgebung eines nominellen Parameterwerts  $p^*$  gültig sind. Die Bestimmung der Ableitungen gliedert sich in den Themenkomplex *Sensitivitätsanalyse* ein, der in [SCS00] in die beiden wesentlichen Gruppen der lokalen und globalen Sensitivitätsanalyse aufgeteilt wird. Die Autoren verstehen unter Sensitivitätsanalyse die Bestimmung der Information, wie die Ausgangsgrößen eines Modells qualitativ oder quantitativ von den Eingangsgrößen abhängen. Wir beschränken uns auf die lokale Sensitivitätsanalyse, die äquivalent zur Berechnung der Ableitungen der Lösung nach Systemparametern ist. Daher meinen wir in der Arbeit mit Sensitivitätsanalyse immer die lokale Variante. Die Ableitungen nennen wir dann folglich *Parametersensitivitäten* oder kurz Sensitivitäten.

Die Herausforderung liegt darin, für hybride Systeme zu analysieren, ob die Parametersensitivitäten existieren und wie sie effizient und numerisch stabil zu berechnen sind – insbesondere auch vor dem Hintergrund der Anwendbarkeit auf automatisch erzeugte multidisziplinäre Systemmodelle. Da die numerischen Aspekte in Kapitel 5 behandelt werden, konzentrieren wir uns in Abschnitt 4.1 auf die analytischen Eigenschaften der Parametersensitivitäten. In Kapitel 3 sind bereits einige Fälle diskutiert worden, in denen für die Lösung des dynamischen Systems übliche Voraussetzungen abgeschwächt wurden. Welche davon für die Sensitivitätsanalyse aufrecht erhalten werden können, zeigt Abschnitt 4.2.

### 4.1 Ableitungen der Lösung nach Parametern

Basierend auf den Sensitivitätsgleichungen glatter Systeme stellen wir in den nächsten Abschnitten vor, wie Sensitivitätsinformationen hybrider und unstetiger Systeme unter bestimmten Voraussetzungen gewonnen werden können. Besonderes Augenmerk wird dabei auf die Situation an Ereignispunkten gelegt. Neben den Sensitivitätsgleichungen gibt es die Möglichkeit, Ableitungen über adjungierte Gleichungen zu berechnen. Diese Vorgehensweise zeigen wir für hybride Systeme, bevor im vierten Abschnitt die Sensitivitätsanalyse für Systeme mit Filippov-Lösung explizit dargelegt wird.

### 4.1.1 Sensitivitätsgleichungen glatter Systeme

Neben der Lösung eines Anfangswertproblems für ein System gewöhnlicher Differentialgleichungen stellt auch die Ableitung dieser Lösung nach Parametern eine zeitabhängige Funktion dar. Die Ableitung gibt die Empfindlichkeit der Lösung gegenüber kleinen Störungen in den Parameterwerten an jedem Zeitpunkt des Lösungsintervalls wieder. Für glatte Systeme definieren wir die zu untersuchenden Problemklassen und klären die Existenz der Parametersensitivitäten.

**Definition 4.1 (Parameterabhängiges Anfangswertproblem)**

Es seien glatte Abbildungen  $f^0 : \mathbb{R}^{n_{x^0}+n_p+1} \rightarrow \mathbb{R}^{n_{x^0}}$  sowie  $A^0 : \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n_{x^0}}$  gegeben. Das System gewöhnlicher Differentialgleichungen

$$\dot{x}^0 = f^0(x^0, p, t) \quad (4.1)$$

mit den Anfangswerten  $x^0(p, t^0) = A^0(p)$  heißt dann *parameterabhängiges Anfangswertproblem*.  $\square$

Wegen der Existenz eines lokalen Flusses [Kön97, Kapitel 4.6] für das erweiterte System

$$\begin{pmatrix} \dot{x}^0 \\ \dot{p} \\ \dot{t} \end{pmatrix} = \begin{pmatrix} f^0(x^0, p, t) \\ 0 \\ 1 \end{pmatrix}$$

gibt es zu jedem  $p$  eine Umgebung und ein Intervall um  $t^0$ , so dass die Lösung  $x^0 = x^0(p, t)$  auf dem Produkt existiert. Wir nehmen an, dass dieses Intervall ein vorgegebenes Intervall  $[t^0, t^\infty]$  enthält.

**Definition 4.2 (Parametersensitivitäten)**

Existiert für ein parameterabhängiges Anfangswertproblem zu jedem  $t \in [t^0, t^\infty]$  die Ableitung  $x_p^0(p, t)$ , dann heißen die Komponenten von  $x_p^0$  *Parametersensitivitäten* der Lösung  $x^0$ .  $\square$

**Satz 4.3 (Existenz von Sensitivitäten)**

Für ein parameterabhängiges Anfangswertproblem mit glatten Abbildungen  $f^0$  und  $A^0$  existieren stetige Parametersensitivitäten in einer Umgebung von  $p$ . Sie erfüllen die folgende lineare Differentialgleichung, die auch Sensitivitätsgleichung genannt wird:

$$\dot{x}_p^0 = f_{x^0}^0(x^0, p, t)x_p^0 + f_p^0(x^0, p, t).$$

Für die Anfangswerte von  $x_p^0$  gilt:  $x_p^0(p, t^0) = A_p^0(p)$ .

**Beweis:**

Die Beweise der Theoreme I.14.1, I.14.3 und I.14.4 in [HNW93] zeigen die Aussage des Satzes.  $\blacksquare$

Die Existenz der Sensitivitäten lässt sich auch für ein System mit variablen, parameterabhängigen Anfangszeitpunkten, wie in Abbildung 4.1 gezeigt, nachweisen:

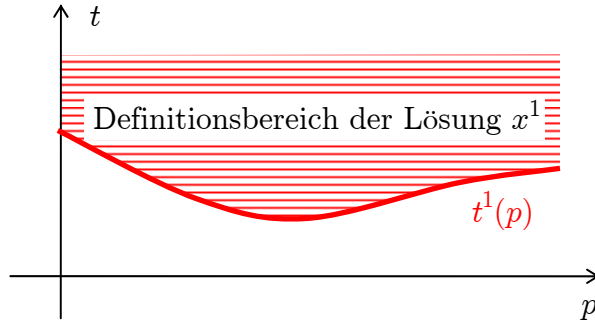


Abbildung 4.1: Definitionsbereich für das parameterabhängige, erweiterte Anfangswertproblem mit parameterabhängiger Anfangszeit  $t^1(p)$ .

**Satz 4.4 (Sensitivitäten für das erweiterte Anfangswertproblem)**

Es sei die glatte Funktion  $t^1 : \mathbb{R}^{n_p} \rightarrow \mathbb{R}$  sowie die glatte Abbildung  $A^1 : \mathbb{R}^{n_p+1} \rightarrow \mathbb{R}^{n_{x^0}}$  gegeben. Das System gewöhnlicher Differentialgleichungen

$$\dot{x}^1 = f^1(x^1, p, t) \quad (4.2)$$

mit den Anfangswerten  $x^1(p, t^1(p)) = A^1(p, t^1(p))$  besitzt dann in einer Umgebung des Punkts  $(p, t^1(p))$  stetige Parametersensitivitäten.

Für die Anfangswerte von  $x_p^1$  gilt:

$$x_p^1(p, t^1(p)) = A_p^1(p, t^1(p)) + \left[ A_t^1(p, t^1(p)) - f^1(x^1(p, t^1(p)), p, t^1(p)) \right] t_p^1(p).$$

**Beweis:**

Das erweiterte Anfangswertproblem wird folgendermaßen umformuliert:

$$\begin{pmatrix} \dot{z} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} f^1(z, p, w) \\ 1 \end{pmatrix}.$$

Als Anfangsbedingung ergibt sich

$$\begin{pmatrix} z(p, 0) \\ w(p, 0) \end{pmatrix} = \begin{pmatrix} A^1(p, t^1(p)) \\ t^1(p) \end{pmatrix}.$$

Auf dieses umformulierte System kann Satz 4.3 angewendet werden. Somit gibt es eine Umgebung um  $p$  und ein Intervall um  $t^1(p)$ , so dass die Lösung  $z$  auf dem Produkt stetige Parametersensitivitäten besitzt. Daher gilt selbiges für die rücktransformierte Lösung  $x^1(p, t) = z(p, t - t^1(p))$ . Die Formel für die Anfangswerte von  $x_p^1$  ergibt sich durch Differenzieren der Identität

$$x^1(p, t) = A^1(p, t^1(p)) + \int_{t^1(p)}^t f^1(x^1(p, \tau), p, \tau) d\tau.$$

■

Satz 4.4 dient als Vorbereitung für den Nachweis, dass hybride Systeme Parametersensitivitäten besitzen. Die parameterabhängigen Anfangszeitpunkte werden bei Systemen mit Schaltfunktionen von Ereigniszeitpunkten erzeugt.

### 4.1.2 Sensitivitätsgleichungen hybrider Systeme

Hybride Systeme setzen sich aus glatten Teilsystemen zusammen, für die jeweils Parametersensitivitäten nach Abschnitt 4.1.1 existieren. Der Übergang am Ereignispunkt, den man auch als Zusammenfügen von Lösungsästen interpretieren kann, spielt die entscheidende Rolle, um die Existenz der Sensitivitäten zu klären.

#### Bemerkung 4.5 (Parameterabhängiges hybrides System)

Für die nachfolgenden Analysen betrachten wir ein hybrides System nach Definition 2.9, das parameterabhängig ist:

$$\dot{x}^0 = f^0(x^0, p, t) \quad \text{für} \quad t^0 \leq t \leq t^1, \quad x^0(p, t^0) = A^0(p), \quad (4.3a)$$

$$y^0 = g^0(x^0, p, t) \quad \text{für} \quad t^0 \leq t \leq t^1, \quad (4.3b)$$

$$\dot{x}^1 = f^1(x^1, p, t) \quad \text{für} \quad t^1 \leq t \leq t^\infty, \quad x^1(p, t^1) = A^1(x^0(p, t^1), p, t^1), \quad (4.3c)$$

$$y^1 = g^1(x^1, p, t) \quad \text{für} \quad t^1 \leq t \leq t^\infty. \quad (4.3d)$$

Die auftretenden Abbildungen  $f^0 : \mathbb{R}^{n_{x^0}+n_p+1} \rightarrow \mathbb{R}^{n_{x^0}}$ ,  $f^1 : \mathbb{R}^{n_{x^1}+n_p+1} \rightarrow \mathbb{R}^{n_{x^1}}$  sowie  $A^0 : \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n_{x^0}}$ ,  $A^1 : \mathbb{R}^{n_{x^0}+n_p+1} \rightarrow \mathbb{R}^{n_{x^1}}$  seien dabei hinreichend glatt. Der Ereigniszeitpunkt  $t^1$  wird eindeutig durch die Gleichung

$$q^1(x^0(p, t^1), p, t^1) = 0$$

mit der glatten Schaltfunktion  $q^1 : \mathbb{R}^{n_{x^0}+n_p+1} \rightarrow \mathbb{R}$  festgelegt.  $\square$

#### Bemerkung 4.6 (Parameterableitungen der algebraischen Variablen)

Gibt es für die Zustände  $x^0$ ,  $x^1$  Parametersensitivitäten, so folgt sofort, dass die algebraischen Größen ebenfalls nach den Parametern differenzierbar sind:

$$y_p^0(p, t) = g_{x^0}^0(x^0(p, t), p, t)x_p^0(x^0(p, t), p, t) + g_p^0(x^0(p, t), p, t) \quad \text{für} \quad t^0 \leq t \leq t^1,$$

$$y_p^1(p, t) = g_{x^1}^1(x^1(p, t), p, t)x_p^1(x^1(p, t), p, t) + g_p^1(x^1(p, t), p, t) \quad \text{für} \quad t^1 \leq t \leq t^\infty.$$

Wir nennen diese Ableitungen ebenfalls Parametersensitivitäten (der algebraischen Variablen).  $\square$

#### Satz 4.7 (Sensitivitäten für hybrides Problem)

Für ein parameterabhängiges hybrides System existieren stetige Parametersensitivitäten in einer Umgebung von  $p^* \in \mathbb{R}^{n_p}$ , falls die Regularitätsbedingung

$$Dq^1(x^0(p^*, t^1), p^*, t^1) := [q_{x^0}^1 f^0 + q_t^1](x^0(p^*, t^1), p^*, t^1) \neq 0 \quad (4.4)$$

zur Schaltzeit  $t^1$  gilt.

#### Beweis:

Die Beweisführung beruht auf dem Beweis von Theorem 1 in [GFB99] bzw. Theorem 3.1

in der Originalarbeit [Fee98]. Mit  $x^0(p, t)$  bzw.  $x^1(p, t)$  sei die Lösung des hybriden Problems bezeichnet, die nach Bemerkung 2.10 existiert. Für das Problem (4.3a) ist Satz 4.3 anwendbar. Für den Übergang bei  $t^1$  definieren wir die Abbildung  $h : U \rightarrow \mathbb{R}^{n_{x^0}+1+n_{x^1}}$ ,

$$h(X, Z, p, T) := \begin{pmatrix} X - x^0(p, t^0) - \int_{t^0}^T f^0(x^0(p, t), p, t) dt \\ q^1(X, p, T) \\ Z - A^1(X, p, T) \end{pmatrix}$$

in einer Umgebung  $U \subset \mathbb{R}^{n_{x^0}+n_{x^1}+n_p+1}$  von  $(X^*, Z^*, p^*, T^*)$  mit  $X^* := x^0(p^*, t^1)$ ,  $T^* := t^1$  und  $Z^* := A^1(X^*, p^*, T^*)$ . Die Abbildung  $h$  ist wohldefiniert, weil  $x^0$  in einer Umgebung von  $(p^*, t^1)$  existiert (Existenz- und Eindeigkeitssatz für gewöhnliche Differentialgleichungen). Es folgt die Identität  $h(X^*, Z^*, p^*, T^*) = 0$ . Die Abbildung  $h$  ist in einer Umgebung der Nullstelle  $(X^*, Z^*, p^*, T^*)$  stetig differenzierbar, daher existiert die quadratische Jacobimatrix  $J := (h_X \ h_Z \ h_T)$  in der Nullstelle. Die Regularität von  $J(X^*, Z^*, p^*, T^*)$  ist äquivalent zur Bedingung  $Dq^1(x^0(p^*, t^1), p^*, t^1) \neq 0$ , siehe Gleichung (4.4). Nach dem Satz über implizite Funktionen gibt es folglich eine Umgebung der Nullstelle, in der stetig differenzierbare Auflösungen  $X(p)$ ,  $Z(p)$  und  $T(p)$  existieren, die  $h = 0$  in einer Umgebung von  $p^*$  erfüllen. Es sind somit die Voraussetzungen von Satz 4.4 gegeben, die die Existenz stetiger Parametersensitivitäten von  $x^1$  in einer Umgebung von  $p^*$  für  $t \geq t^1$  garantieren. ■

#### Bemerkung 4.8 (Sensitivitäten am Schaltzeitpunkt)

Unter den Voraussetzungen von Satz 4.7 können die Parametersensitivitäten  $x_p^1(p, t^1(p))$  in einer Umgebung von  $p^*$  mit Hilfe von Satz 4.4 berechnet werden:

$$\begin{aligned} x_p^1(p, t^1(p)) &= \left[ A_{x^0}^1 x_p^0(p, t^1(p)) + A_p^1 \right] \left( x^0(p, t^1(p)), p, t^1(p) \right) \\ &\quad + \left( \left[ A_{x^0}^1 f^0 + A_t^1 \right] \left( x^0(p, t^1(p)), p, t^1(p) \right) - f^1 \left( x^1(p, t^1(p)), p, t^1(p) \right) \right) t_p^1(p) \end{aligned} \quad (4.5)$$

mit

$$t_p^1(p) = \frac{q_{x^0}^1 \left( x^0(p, t^1(p)), p, t^1(p) \right) x_p^0(p, t^1(p)) + q_p^1 \left( x^0(p, t^1(p)), p, t^1(p) \right)}{-Dq^1 \left( x^0(p, t^1(p)), p, t^1(p) \right)}. \quad (4.6)$$

□

#### Korollar 4.9 (Sensitivitäten für mehrschichtiges Problem)

Gegeben sei ein mehrschichtiges parameterabhängiges System mit mehreren (endlich vielen) eindeutigen Schaltzeitpunkten (siehe Abbildung 4.2), die paarweise voneinander verschiedenen seien und an denen jeweils die Regularitätsbedingung  $Dq^i \neq 0$  erfüllt sei. Dann existieren stetige Parametersensitivitäten in einer Umgebung von  $p^*$ . Falls zusätzlich algebraische Variablen über glatte Abbildungen definiert sind, so existieren auch die Ableitungen der algebraischen Variablen nach  $p$  in der Umgebung von  $p^*$ . ■

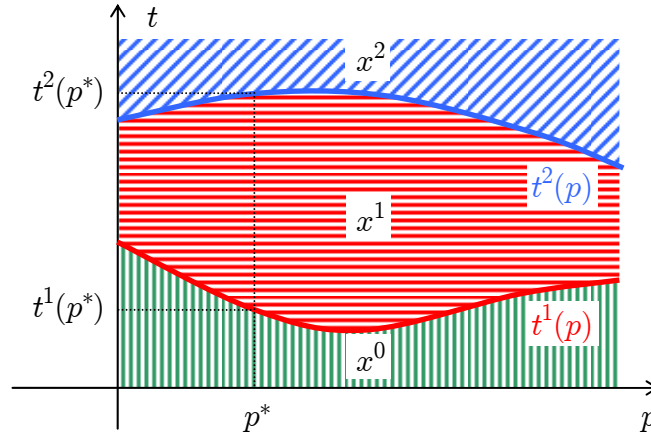


Abbildung 4.2: Definitionsbereiche für ein mehrschichtiges parameterabhängiges hybrides System.

#### Bemerkung 4.10 (Zusammenfügen von Lösungsästen)

Betrachtet man die Lösungsäste  $x^0$  und  $x^1$  unter der Voraussetzung  $n_x := n_{x^0} = n_{x^1}$  als Bestandteile einer Abbildung  $x : \mathbb{R}^{n_p+1} \rightarrow \mathbb{R}^{n_x}$  mit

$$x(p, t) := \begin{cases} x^0(p, t), & \text{falls } t < t^1(p), \\ x^1(p, t), & \text{falls } t \geq t^1(p), \end{cases}$$

dann stellt sich die Frage, ob die zusammengesetzte Lösung  $x$  zur Schaltzeit  $t^1$  als Ganzes nach  $p$  differenzierbar ist. Diese Frage ist bislang mit Satz 4.7 nicht beantwortet und wird im Folgenden untersucht. In [Roz67] und [GFB99] werden hinreichende Bedingungen für die Existenz von  $x_p$  bereits genannt. In Satz 4.14 wird zusätzlich gezeigt, dass diese für gewisse Fälle auch notwendig sind.  $\square$

#### Satz 4.11 (Hinreichende Bedingungen für die Existenz von Sensitivitäten)

Gegeben sei ein hybrides System (4.3), das die Regularitätsbedingung (4.4) erfülle; zudem gelte  $n_{x^0} = n_{x^1}$ . Es sei  $p^* \in \mathbb{R}^{n_p}$  ein fest gewählter Parametersatz. Dann gilt:

a) Es gibt eine Umgebung von  $(p^*, t^1(p^*))$ , in der stetige Sensitivitäten  $x_p$  existieren, wenn

$$\begin{aligned} x^1(p, t^1(p)) &= x^0(p, t^1(p)) && (\text{Stetigkeit von } x), \\ \dot{x}^1(p, t^1(p)) &= \dot{x}^0(p, t^1(p)) && (\text{Stetigkeit von } \dot{x}) \end{aligned}$$

für alle  $p$  aus einer gewissen Umgebung von  $p^*$  gelten.

b) Falls  $x^1(p, t^1(p)) = x^0(p, t^1(p))$  in einer Umgebung von  $p^*$  und  $t_p^1(p^*) = 0$  gelten, so existiert  $x_p(p^*, t^1(p^*))$ .

c) Falls  $t^1$  bei  $p^*$  lokal konstant ist, dann existiert  $x_p$  in einer Umgebung von  $(p^*, t^1(p^*))$ .

**Beweis:**

a) Nach Satz 4.7 existieren die Ableitungen  $x_p^0$  und  $x_p^1$  in einer Umgebung von  $(p^*, t^1(p^*))$ . Ohne Einschränkung nehmen wir  $n_p = 1$  an. Es gilt:

$$x(p, t) = A^0(p) + \int_{t^0}^{\min(t, t^1(p))} \dot{x}^0(p, \tau) d\tau + \left[ x^1(p, t^1(p)) - x^0(p, t^1(p)) + \int_{t^1(p)}^t \dot{x}^1(p, \tau) d\tau \right] \cdot \chi_{\{t \geq t^1(p)\}}(t)$$

mit der charakteristischen Funktion  $\chi_D(t) := \begin{cases} 1, & \text{falls } t \in D, \\ 0, & \text{falls } t \notin D. \end{cases}$

Somit folgt nach der Voraussetzung  $x^1(p, t^1(p)) = x^0(p, t^1(p))$  die Gleichung

$$\frac{x(p + \Delta p, t^1(p)) - x(p, t^1(p))}{\Delta p} = B + C$$

mit

$$B := \frac{A^0(p + \Delta p) - A^0(p)}{\Delta p} + \int_{t^0}^{\min(t^1(p), t^1(p + \Delta p))} \frac{\dot{x}^0(p + \Delta p, \tau) - \dot{x}^0(p, \tau)}{\Delta p} d\tau$$

und

$$C := -\frac{1}{\Delta p} \int_{\min(t^1(p), t^1(p + \Delta p))}^{t^1(p)} \dot{x}^0(p, \tau) d\tau + \frac{1}{\Delta p} \int_{t^1(p + \Delta p)}^{t^1(p)} \dot{x}^1(p + \Delta p, \tau) d\tau \cdot \chi_{\{t \geq t^1(p + \Delta p)\}}(t^1(p)).$$

Es gilt:

$$x_p^0(p, t^1(p)) = A_p^0(p) + \int_{t^0}^{t^1(p)} \dot{x}_p^0(p, \tau) d\tau.$$

Somit folgt:  $B \rightarrow x_p^0(p, t^1(p))$  für  $\Delta p \rightarrow 0$ , da die Minimumfunktion stetig und der Integrand bzgl.  $\Delta p$  stetig fortsetzbar ist. Es gilt weiter:

$$\|C\| \leq \left\| \frac{1}{\Delta p} \int_{t^1(p)}^{t^1(p + \Delta p)} (\dot{x}^1(p + \Delta p, \tau) - \dot{x}^0(p, \tau)) d\tau \right\|.$$

Mit Lemma 4.12 und dem Grenzübergang  $\Delta p \rightarrow 0$  folgt dann:

$$\|C\| \leq |t_p^1(p)| \cdot \|\dot{x}^1(p, t^1(p)) - \dot{x}^0(p, t^1(p))\|.$$



Nach Voraussetzung gilt  $\dot{x}^1(p, t^1(p)) = \dot{x}^0(p, t^1(p))$ , damit ergibt sich  $C = 0$ . Differenzieren von  $x^1(p, t^1(p)) = x^0(p, t^1(p))$  nach  $p$  ergibt wegen der Voraussetzung, dass die zeitlichen Ableitungen gleich sind:  $x_p^0(p, t(p)) = x_p^1(p, t(p))$ . Somit haben wir  $x_p(p, t(p)) = x_p^0(p, t(p)) = x_p^1(p, t(p))$  bewiesen.

b) Für  $p = p^*$  kann der Ausdruck  $\|C\|$  wie im Beweisteil a) abgeschätzt werden. Da  $t_p^1(p^*) = 0$  gilt, folgt  $C = 0$  und damit die Existenz von  $x_p(p^*, t^1(p^*)) = x_p^0(p^*, t^1(p^*)) = x_p^1(p^*, t^1(p^*))$ .

c) Da  $t^1$  lokal konstant ist, folgt aus der Definition von  $x$  in einer Umgebung von  $(p^*, t^*)$ :

$$x_p(p, t) = \begin{cases} x_p^0(p, t), & \text{falls } t < t^1(p^*), \\ x_p^1(p, t), & \text{falls } t \geq t^1(p^*). \end{cases}$$

Also gilt die Gleichung  $x_p(p, t^1(p)) = x_p^1(p, t^1(p))$ . ■

### Lemma 4.12 (Parameterabhängiges Integral)

Es sei  $z : U_1 \times U_2 \rightarrow \mathbb{R}^{n_z}$ ,  $(\Delta p, \tau) \mapsto z(\Delta p, \tau)$  eine stetig differenzierbare Funktion mit Umgebungen  $U_1, U_2 \subset \mathbb{R}$  von  $0, a \in \mathbb{R}$ . Weiter sei  $b : U_1 \rightarrow \mathbb{R}$ ,  $r \mapsto b(r)$  eine stetig differenzierbare Funktion. Es gelte:  $b(0) = a$ . Wir definieren damit die neue Funktion  $G : U_1 \times U_1 \rightarrow \mathbb{R}^{n_z}$ ,

$$G(\Delta p, r) := \int_a^{b(r)} z(\Delta p, \tau) d\tau.$$

Dann ist die Funktion  $H(\Delta p) := G(\Delta p, \Delta p)$  stetig differenzierbar auf  $U_1$  und es gilt:

$$\lim_{\Delta p \rightarrow 0} \frac{1}{\Delta p} H(\Delta p) = z(0, a) b_r(0).$$

#### Beweis:

Nach dem Differentiationssatz für parameterabhängige Integrale [Kön97, Kapitel 8.4] ist  $G$  stetig differenzierbar, somit gilt das auch für  $H$ . Es folgt wegen  $H(0) = 0$ :

$$\lim_{\Delta p \rightarrow 0} \frac{1}{\Delta p} H(\Delta p) = H_{\Delta p}(0) = G_{\Delta p}(0, 0) + G_r(0, 0) = z(0, a) b_r(0). \quad \blacksquare$$

### Bemerkung 4.13 (Glattheit von $f$ )

Unter den Voraussetzungen von Satz 4.11 a) ist die Funktion

$$f(x, p, t) := \begin{cases} f^0(x, p, t), & \text{falls } t < t^1(p), \\ f^1(x, p, t), & \text{falls } t \geq t^1(p) \end{cases}$$

stetig, aber nicht notwendigerweise differenzierbar. Daher sind diese Voraussetzungen schwächer als die im Beweis von Satz 4.3, in dem mindestens die stetige Differenzierbarkeit von  $f$  gefordert wird. Damit kann Satz 4.11 nicht direkt über Satz 4.3 begründet werden. Stetige Differenzierbarkeit von  $f$  würde auch die zweifach stetige Differenzierbarkeit von  $x$  implizieren, die unter den Voraussetzungen von Satz 4.11 a) im Allgemeinen nicht gegeben ist. □

**Satz 4.14 (Notwendige Bedingungen für die Existenz von Sensitivitäten)**

Für  $n_p = 1$  sei ein hybrides System (4.3) mit glatten Abbildungen gegeben, das die Regularitätsbedingung (4.4) erfüllt; zudem gelte  $n_{x^0} = n_{x^1}$ . Es sei vorausgesetzt, dass  $x_p$  in einer Umgebung von  $(p^*, t^1(p^*))$  existiert. Dann gilt:

a) Wenn  $t_p^1(p^*) \neq 0$  erfüllt ist, dann gibt es eine Umgebung von  $p^*$ , in der folgende Identitäten gelten:

$$\begin{aligned} x^1(p, t^1(p)) &= x^0(p, t^1(p)), \\ \dot{x}^1(p, t^1(p)) &= \dot{x}^0(p, t^1(p)). \end{aligned}$$

b) Falls in jeder Umgebung von  $p^*$  ein  $p$  mit  $t_p^1(p) \neq 0$  existiert, dann folgt:

$$\begin{aligned} x^1(p^*, t^1(p^*)) &= x^0(p^*, t^1(p^*)), \\ \dot{x}^1(p^*, t^1(p^*)) &= \dot{x}^0(p^*, t^1(p^*)). \end{aligned}$$

**Beweis:**

a) Da  $t_p^1 \neq 0$  in einer Umgebung von  $p^*$  gilt, folgt in dieser die Gleichung:

$$x_p^0(p, t^1(p)) = x_p^1(p, t^1(p)). \quad (*)$$

Da  $x_p$  in einer Umgebung von  $(p^*, t^1(p^*))$  existiert, ist  $x$  dort insbesondere stetig. Also folgt:  $x^1(p, t^1(p)) = x^0(p, t^1(p))$ . Mit Hilfe der Regularitätsbedingung lässt sich diese Gleichung nach  $p$  differenzieren. Man erhält unter Ausnutzung von (\*):

$$\left( \dot{x}^0(p, t^1(p)) - \dot{x}^1(p, t^1(p)) \right) t_p^1(p) = 0,$$

was die Identität  $\dot{x}^1(p, t^1(p)) = \dot{x}^0(p, t^1(p))$  zur Folge hat.

b) Es sei  $(p^k)_{k \in \mathbb{N}}$ ,  $p^k \rightarrow p^*$  für  $k \rightarrow \infty$ , eine Folge mit  $t_p^1(p^k) \neq 0$ . Dann folgt aus a):

$$\begin{aligned} x^1(p^k, t^1(p^k)) &= x^0(p^k, t^1(p^k)), \\ \dot{x}^1(p^k, t^1(p^k)) &= \dot{x}^0(p^k, t^1(p^k)). \end{aligned}$$

Der Grenzübergang  $k \rightarrow \infty$  beweist wegen der Stetigkeit der beteiligten Abbildungen die Behauptung. ■

**Korollar 4.15 (Sensitivitäten am Ereignispunkt)**

Für  $n_p = 1$  sei ein hybrides System (4.3) mit glatten Abbildungen gegeben, das die Regularitätsbedingung (4.4) erfüllt; zudem gelte  $n_{x^0} = n_{x^1}$ . Weiter sei vorausgesetzt, dass  $t_p^1(p^*) \neq 0$  gilt. Die Sensitivitäten  $x_p$  existieren *genau dann* in einer Umgebung des Punkts  $(p^*, t^1(p^*))$ , wenn in einer Umgebung folgendes gilt:

$$\begin{aligned} x^1(p, t^1(p)) &= x^0(p, t^1(p)) && \text{(Stetigkeit im Zustand } x), \\ \dot{x}^1(p, t^1(p)) &= \dot{x}^0(p, t^1(p)) && \text{(Stetigkeit in den Zeitableitungen } \dot{x}). \end{aligned}$$

■

Die Voraussetzung  $n_p = 1$  in Satz 4.14 und Korollar 4.15 ist deswegen sinnvoll, weil für jede Komponente eines Parametervektors unabhängige Aussagen bzgl. der Existenz von Sensitivitäten nötig sind.

### Beispiel 4.16 (Unstetige Sensitivitäten)

Wir lösen das eindimensionale System

$$\dot{x}^0 = \begin{cases} p, & \text{falls } t < 3, \\ 0, & \text{falls } t > 3. \end{cases}, \quad x^0(p, 0) = 0, \quad A^1(x^0, p, t) = 2x^0.$$

Mit  $q^1(x^0, p, t) = t - 3$  gilt  $t^1(p) = 3 = \text{const.}$  Damit existieren nach Satz 4.11 c) die Sensitivitäten der zusammengesetzten Lösung

$$x(p, t) = \begin{cases} pt, & \text{falls } t < 3, \\ 6p, & \text{falls } t \geq 3. \end{cases} \quad \text{mit} \quad x_p(p, t) = \begin{cases} t, & \text{falls } t < 3, \\ 6, & \text{falls } t \geq 3. \end{cases}$$

Hierbei ist zu beobachten, dass  $x_p$  zum Schaltzeitpunkt  $t^1 = 3$  unstetig in der Zeit ist.  $\square$

### Beispiel 4.17 (Unstetige Lösung)

Das System

$$\dot{x}^0 = p, \quad x^0(p, 0) = 0, \quad q^1(x^0, p, t) = x^0 - 3, \quad A^1(x^0, p, t) = 2x^0$$

besitzt für  $p > 0$  die Lösung

$$x(p, t) = \begin{cases} x^0(p, t) = pt, & \text{falls } 0 \leq t < \frac{3}{p}, \\ x^1(p, t) = pt + 3, & \text{falls } \frac{3}{p} \leq t. \end{cases}$$

Da  $t^1(p) = \frac{3}{p}$  gilt, folgt  $t_p^1 \neq 0$  und  $x_p^0(p, t^1(p)) = x_p^1(p, t^1(p)) = \frac{3}{p}$ . Die zusammengesetzte Funktion  $x$  ist aber unstetig bzgl.  $p$ , somit nicht differenzierbar, vgl. Korollar 4.15. Die Bedingung  $x_p^0(p, t^1(p)) = x_p^1(p, t^1(p))$  garantiert also nicht, dass die Sensitivitäten  $x_p$  der zusammengesetzten Lösung existieren.  $\square$

Beispiel 4.17 zeigt ein einfaches System, für das die Sensitivitäten der einzelnen Lösungsäste  $x^0, x^1$  nach Satz 4.7 existieren – aber nicht die der zusammengesetzten Funktion  $x$ . Wenn die Sensitivitätswerte nur an den Zeitpunkten interessieren, die keine Ereigniszeitpunkte sind bzw. wenn die Lösungsäste nicht als Bestandteile einer gemeinsamen Lösung betrachtet werden, dann reichen die Voraussetzungen von Satz 4.7 aus.

## 4.1.3 Adjungierte Gleichungen hybrider Systeme

In manchen Anwendungen werden die Parametersensitivitäten nur von wenigen algebraischen Variablen am Endzeitpunkt des Zeitintervalls benötigt. Für diese Fälle ist es vorteilhaft, anstatt der Sensitivitätsgleichungen die *adjungierten Gleichungen* zu lösen. Ursprünglich kommt dieser Ansatz aus der Optimalsteuerung (z. B. [HA97]). Für glatte Systeme ist die Herleitung der adjungierten Gleichungen in [CLPS03] zu finden. Für

Systeme mit unstetiger rechter Seite und unstetigen Zustandsänderungen ergeben sich Sprungbedingungen für die adjungierten Variablen am Ereignispunkt [GM80]. Nachfolgend führen wir die Berechnungen für die mehrschichtigen hybriden Systeme aus. An den Ereignispunkten werden – anders als für die Zustandssensitivitäten – die Informationen der nachfolgenden Schicht auf die aktuelle Schicht transferiert, was typisch für die Methode der adjungierten Variablen ist.

**Definition 4.18 (Adjungierte Gleichungen)**

Wir betrachten für  $m \geq 0$  ein  $(m + 1)$ -schichtiges parameterabhängiges System entsprechend Bemerkung 2.11 und analog zu System (4.3) mit einer eindeutigen Lösung. Wir definieren die linearen Differentialgleichungssysteme

$$\dot{\lambda}_i = -f_{x^i}^i(x^i, p, t)\lambda_i$$

für  $i = 0, 1, \dots, m$  und  $t \in [t^i, t^{i+1}]$  mit  $t^i = t^i(p)$  für  $i = 1, \dots, m$ . Der Zeitpunkt  $t^{m+1} := t^\infty > t^m$  sei dabei fest gewählt und *nicht* aus der Nullstelle einer Schaltfunktion  $q^{m+1}$  berechnet. Die Funktionen  $\lambda_i : [t^i, t^{i+1}] \rightarrow \mathbb{R}^{n_{x^i}}$  werden *adjungierte Variablen* genannt, die definierenden Differentialgleichungen *adjungierte Gleichungen*.  $\square$

**Bemerkung 4.19 (Existenz der adjungierten Variablen)**

Da die adjungierten Gleichungen linear sind, existieren die adjungierten Variablen auf den vorgegebenen Zeitintervallen zu beliebig gewählten Anfangswerten  $\lambda_i(t^i)$  [Kön97, Kapitel 4.3]. Ebenso können alternativ die Werte für  $\lambda_i$  an den Endzeitpunkten  $t^{i+1}$  ( $i = 0, \dots, m$ ) beliebig gewählt werden.  $\square$

**Lemma 4.20 (Umformulierte Sensitivitäten am Ereignispunkt)**

Für ein  $(m + 1)$ -schichtiges parameterabhängiges System (4.3), das an den Schaltzeitpunkten jeweils die Regularitätsbedingung  $Dq^i(x^{i-1}(p, t^i), p, t^i) \neq 0$  (siehe Bedingung (4.4)) erfüllt, gilt mit den Matrizen

$$\begin{aligned} D_i &:= \left[ A_{x^{i-1}}^i + \left( f^i(x^i(p, t^i), p, t^i) - A_{x^{i-1}}^i f^{i-1} - A_t^i \right) (Dq^i)^{-1} q_{x^{i-1}}^i \right] (x^{i-1}(p, t^i), p, t^i), \\ E_i &:= \left[ A_p^i + \left( f^i(x^i(p, t^i), p, t^i) - A_{x^{i-1}}^i f^{i-1} - A_t^i \right) (Dq^i)^{-1} q_p^i \right] (x^{i-1}(p, t^i), p, t^i) \end{aligned}$$

an den Ereignispunkten die folgende Beziehung zwischen den Sensitivitäten:

$$x_p^i(p, t^i) = D_i x_p^{i-1}(p, t^i) + E_i \quad (i = 1, \dots, m).$$

**Beweis:**

Die Formeln ergeben sich aus einer Umformulierung der Gleichungen von Bemerkung 4.8.  $\blacksquare$

**Satz 4.21 (Sensitivitäten mit adjungierten Gleichungen)**

Wir betrachten für  $m \geq 0$  ein  $(m + 1)$ -schichtiges parameterabhängiges System (4.3) zusammen mit einer glatten Funktion  $c : \mathbb{R}^{n_{x^m} + n_p + 1} \rightarrow \mathbb{R}$ . An den Schaltzeitpunkten sei jeweils die Regularitätsbedingung (4.4) erfüllt. Des Weiteren sei  $t^{m+1}$  mit  $t^{m+1} > t^m$  ein fest vorgegebener Zeitpunkt.

Zu einer Zielfunktion  $C(p) := c(x^m(p, t^{m+1}), p, t^{m+1})$  kann dann die Ableitung  $C_p(p^*)$  mit Hilfe der adjungierten Variablen wie folgt bestimmt werden:

$$\begin{aligned} C_p(p^*) &= c_p(x^m(p^*, t^{m+1}), p^*, t^{m+1}) + \sum_{i=0}^m \int_{t^i}^{t^{i+1}} \lambda_i^\top(t) f_p^i(x^i(p^*, t), p^*, t) dt \\ &\quad + \lambda_0^\top(t^0) A_p^0(p^*) + \sum_{i=1}^m \lambda_i^\top(t^i) E_i. \end{aligned}$$

Als Endwerte für die adjungierten Variablen seien dabei

$$\lambda_m(t^{m+1}) := c_{x^m}(x^m(p^*, t^{m+1}), p^*, t^{m+1})^\top \quad \text{und} \quad \lambda_{i-1}(t^i) := D_i^\top \lambda_i(t^i)$$

für  $i = 1, \dots, m$  definiert.

**Beweis:**

Wegen  $\dot{x}^i(p, t) = f^i(x^i(p, t), p, t)$  auf  $[t^i(p), t^{i+1}(p)]$  gilt:

$$C(p) = c(x^m(p, t^{m+1}), p, t^{m+1}) + \sum_{i=0}^m \int_{t^i(p)}^{t^{i+1}(p)} \lambda_i^\top \left( f^i(x^i(p, t), p, t) - \dot{x}^i(p, t) \right) dt.$$

Somit folgt (zur besseren Lesbarkeit z. T. ohne Argumente):

$$\begin{aligned} C_p(p^*) &= [c_{x^m} x_p^m + c_p](t^{m+1}) + \sum_{i=0}^m \int_{t^i(p^*)}^{t^{i+1}(p^*)} \lambda_i^\top (f_{x^i}^i x_p^i + f_p^i - \dot{x}_p^i) dt \\ &\quad + \sum_{i=0}^m \left[ \lambda_i^\top(t^{i+1}) \underbrace{(f^i - \dot{x}^i)(t^{i+1})}_{=0} t_p^{i+1} - \lambda_i^\top(t^i) \underbrace{(f^i - \dot{x}^i)(t^i)}_{=0} t_p^i \right]. \end{aligned}$$

Die partielle Integration

$$\int_{t^i}^{t^{i+1}} \lambda_i^\top \dot{x}_p^i dt = \lambda_i^\top x_p^i \Big|_{t^i}^{t^{i+1}} - \int_{t^i}^{t^{i+1}} \dot{\lambda}_i^\top x_p^i dt$$

liefert dann:

$$C_p(p^*) = [c_{x^m} x_p^m + c_p](t^{m+1}) - \sum_{i=0}^m \lambda_i^\top x_p^i \Big|_{t^i}^{t^{i+1}} + \sum_{i=0}^m \int_{t^i}^{t^{i+1}} \underbrace{[(\lambda_i^\top f_{x^i}^i + \dot{\lambda}_i^\top) x_p^i + \lambda_i^\top f_p^i]}_{=0, \text{ vgl. Def. 4.18}} dt.$$

Da  $\lambda_i$  die adjungierten Gleichungen erfüllt, reduzieren sich die Integrale zu:

$$\begin{aligned} C_p(p^*) &= c_p(t^{m+1}) + (c_{x^m} - \lambda_m^\top) x_p^m(t^{m+1}) + \sum_{i=1}^m [\lambda_i^\top x_p^i(t^i) - \lambda_{i-1}^\top x_p^{i-1}(t^i)] + \lambda_0^\top x_p^0(t^0) \\ &\quad + \sum_{i=0}^m \int_{t^i}^{t^{i+1}} \lambda_i^\top f_p^i dt. \end{aligned}$$

Die Identität aus Lemma 4.20 ergibt

$$\lambda_i^\top x_p^i(t^i) - \lambda_{i-1}^\top x_p^{i-1}(t^i) = \underbrace{(\lambda_i^\top D_i - \lambda_{i-1}^\top)(t^i)}_{=0 \text{ nach Voraussetzung}} x_p^{i-1}(t^i) + \lambda_i^\top(t^i) E_i,$$

woraus insgesamt mit den Endwerten von  $\lambda_{i-1}(t^i)$  die Behauptung bewiesen ist. ■

#### 4.1.4 Parametersensitivitäten der Filippov-Lösung

Die Aussagen aus den vergangenen Abschnitten 4.1.2 und 4.1.3 gelten insbesondere für parameterabhängige Systeme mit Filippov-Lösung gemäß Abschnitt 3.1.2, weil sie als hybride Systeme (4.3) darstellbar sind. Da sich nach Bemerkung 3.15 die Probleme aus Tabelle 3.4 mit *nichtglatten* Schaltfunktionen als hybride Systeme schreiben lassen, gelten auch für diese Probleme die Aussagen der letzten Abschnitte. Hier wird die Größe der Problemklasse der hybriden Systeme eindrucksvoll deutlich. Eine parameterabhängige Schaltfunktion ist dabei regulär, wenn sie die analogen Eigenschaften aus Definition 3.18 für die um die Parameterdimension erweiterte Schaltfläche  $\Sigma := \{(x, p, t) : q(x, p, t) = 0\}$  erfüllt. Um die Feinheiten der Sensitivitätsberechnung der genannten Systeme zu erkennen, werden im Folgenden zwei Fälle im Detail diskutiert.

##### Beispiel 4.22 (Unstetige rechte Seite und reguläre Schaltfunktion)

Das parameterabhängige System mit unstetiger rechter Seite und regulärer Schaltfunktion

$$\dot{x} = \begin{cases} f^+(x, p, t), & \text{falls } q^+(x, p, t) > 0, \\ f^-(x, p, t), & \text{falls } q^-(x, p, t) < 0 \end{cases}$$

habe zur Zeit  $t^0$  den Anfangswert  $A(p)$ . Es sei  $q^+(A(p^*), p^*, t^0) > 0$ . Damit gilt für das entsprechende hybride System

$$\dot{x}^0 = f^+(x^0, p^*, t) \text{ für } t \in [t^0, t^1]$$

mit  $q^+(x^0(p^*, t^1), p^*, t^1) = 0$ . Unter den Annahmen

$$D^+q^+(x^0(p^*, t^1), p^*, t^1) < 0 \text{ und } D^-q^+(x^0(p^*, t^1), p^*, t^1) > 0$$

folgt die Phase mit Filippov-Lösung:

$$\dot{x}^1 = f^*(x^1, p^*, t) = \left[ \frac{D^+q^+ f^- - D^-q^+ f^+}{D^+q^+ - D^-q^+} \right] (x^1, p^*, t) \text{ für } t \in [t^1, t^2].$$

Der Ereigniszeitpunkt  $t^2$  sei durch die Gleichung  $D^-q^+(x^1(p^*, t^2), p^*, t^2) = 0$  definiert, die der Lösung das Verlassen der Schaltfläche ermöglicht:

$$\dot{x}^2 = f^-(x^2, p^*, t) \text{ für } t \in [t^2, t^\infty].$$

Die Sensitivitäten erfüllen dann die folgenden Sensitivitätsgleichungen:

$$\begin{aligned} \dot{x}_p^0 &= f_x^+(x^0, p^*, t) x_p^0 + f_p^+(x^0, p^*, t), & t \in [t^0, t^1], \\ \dot{x}_p^1 &= f_x^*(x^1, p^*, t) x_p^1 + f_p^*(x^1, p^*, t), & t \in [t^1, t^2], \\ \dot{x}_p^2 &= f_x^-(x^2, p^*, t) x_p^2 + f_p^-(x^2, p^*, t), & t \in [t^2, t^\infty] \end{aligned}$$

mit den Anfangswerten

$$\begin{aligned} x_p^0(p^*, t^0) &= A_p(p^*), \\ x_p^1(p^*, t^1) &= x_p^0(p^*, t^1) + \left[ \frac{(f^+ - f^-)(q_x^+ x_p^0(p^*, t^1) + q_p^+)}{q_x^+(f^- - f^+)} \right] (x^0(p^*, t^1), p^*, t^1), \\ x_p^2(p^*, t^2) &= x_p^1(p^*, t^2). \end{aligned}$$

Jeder Übergang einer Filippov-Lösung in eine klassische Lösung, der mit  $D^-q = 0$  (analog  $D^+q = 0$ ) angezeigt wird, zieht die Identität  $f^* = f^-$  am Ereignispunkt nach sich. Deswegen gehen die Sensitivitäten  $x_p^1$  und  $x_p^2$  stetig in der Zeit ineinander über.  $\square$

### Beispiel 4.23 (Unstetige Zustandsänderung und Filippov-Lösung)

Zur Zeit  $t^0$  habe ein hybrides System den Anfangswert  $A^0(p^*)$ . Es gelte

$$\dot{x}^0 = f^0(x^0, p^*, t) \text{ für } t \in [t^0, t^1]$$

mit  $q^1(x^0(p^*, t^1), p^*, t^1) = 0$ . Es sei

$$x^1(p^*, t^1) = A^1(x^0(p^*, t^1), p^*, t^1) \text{ und } \rho^1(x^1(p^*, t^1), p^*, t^1) = 0.$$

Mit dem parameterabhängigen System mit unstetiger rechter Seite

$$\dot{x}^1 = \begin{cases} f^+(x^1, p, t), & \rho^1(x^1, p, t) > 0, \\ f^-(x^1, p, t), & \rho^1(x^1, p, t) < 0 \end{cases}$$

nehmen wir  $D^+\rho^1(x^1(p^*, t^1), p^*, t^1) < 0$  und  $D^-\rho^1(x^1(p^*, t^1), p^*, t^1) > 0$  an. Die resultierende Filippov-Lösung errechnet sich aus

$$\dot{x}^1 = f^*(x^1, p^*, t) = \left[ \frac{D^+\rho^1 f^- - D^-\rho^1 f^+}{D^+\rho^1 - D^-\rho^1} \right] (x^1, p^*, t) \text{ für } t > t^1.$$

Die Sensitivitäten berechnen sich gemäß Gleichung (4.5) wie folgt:

$$\begin{aligned} \dot{x}_p^0 &= f_x^0(x^0, p^*, t)x_p^0 + f_p^0(x^0, p^*, t), & t \in [t^0, t^1], \\ \dot{x}_p^1 &= f_x^*(x^1, p^*, t)x_p^1 + f_p^*(x^1, p^*, t), & t \in [t^1, t^2]. \end{aligned}$$

Als Anfangswerte dienen dabei die Werte

$$\begin{aligned} x_p^0(p^*, t^0) &= A_p^0(p^*), \\ x_p^1(p^*, t^1) &= [A_{x^0}^1(x_p^0(p^*, t^1) + f^0 t_p^1) + A_p^1] (x^0(p^*, t^1), p^*, t^1) \\ &\quad + \left( A_t^1(x^0(p^*, t^1), p^*, t^1) - f^*(x^1(p^*, t^1), p^*, t^1) \right) t_p^1, \\ t_p^1 &= - \left[ \frac{q_{x^0}^1 x_p^0(p^*, t^1) + q_p^1}{q_{x^0}^1 f^0 + q_t^1} \right] (x^0(p^*, t^1), p^*, t^1). \end{aligned}$$

In die Berechnung von  $x_p^1(p^*, t^1)$  gehen also die Schaltfunktionen  $q^1$  (über  $t^1$  sowie  $t_p^1$ ) und  $\rho^1$  (über  $f^*$ ) ein.  $\square$

In der Literatur werden die Sensitivitäten von Systemen mit unstetigen rechten Seiten (inkl. Filippov-Lösung) bereits in [Fil88, §11] für die verschiedensten Fälle berechnet. Auch in [SA05] sind Formeln für die Sensitivitätsgleichungen der Filippov-Lösung und die jeweiligen Anfangswerte der Lösungsphasen zu finden.

Bisher ist die Übertragung der Sensitivitätsanalyse auf hybride Systeme ohne relevante Zusatzvoraussetzungen im Vergleich zur alleinigen Lösung der Systeme möglich gewesen. Diese Aussage gilt – wie wir soeben gesehen haben – auch für einen Teil der Besonderheiten, die von automatisch erzeugten Systemmodellen hervorgerufen werden, vgl. Abschnitt 3.2.1. Die Problematik von gleichzeitig aktiven Schaltfunktionen lässt sich nicht mehr so einfach im Rahmen der aus der Literatur bekannten Ergebnisse behandeln. Der gesamte nächste Abschnitt ist daher diesem Thema gewidmet.

## 4.2 Gleichzeitig aktive Schaltfunktionen

Wie für die Lösung von Systemen mit Schaltfunktionen wird auch für die Sensitivitätsanalyse in der Literatur [Boc87, GFB99, Cal00] vorausgesetzt, dass am Ereigniszeitpunkt genau eine Schaltfunktion aktiv ist. In Abschnitt 3.2.2 werden bereits Systeme analysiert, die am Ereigniszeitpunkt gleichzeitig mehrere aktive Schaltfunktionen besitzen. Dort liegt der Schwerpunkt darauf, eine rechte Seite definieren zu können, die den physikalischen Eigenschaften des modellierten Systems entspricht. Im ersten Teil des vorliegenden Abschnitts werden hinreichende Bedingungen formuliert, unter denen Parametersensitivitäten für Systeme mit mehreren aktiven Schaltfunktionen existieren, worauf im zweiten Teil folgend Beispiele das Lösungskonzept verdeutlichen sollen.

### 4.2.1 Problemanalyse

Wir beschäftigen uns nun mit der Fragestellung, unter welchen Voraussetzungen die Sensitivitäten für hybride Systeme mit mehreren aktiven Schaltfunktionen existieren und wie sie sich gegebenenfalls berechnen lassen. Es gelte für den Vektor der aktiven Schaltfunktionen  $q = (q^1, \dots, q^{n_q})^\top : \mathbb{R}^{n_x+n_p+1} \rightarrow \mathbb{R}^{n_q}$  die Identität  $q(x^0(p^*, t^*), p^*, t^*) = 0$ . Da für jeden Parameter unabhängige Untersuchungen notwendig sind, nehmen wir  $n_p = 1$  an.

#### Bemerkung 4.24 (Lösungsstruktur bei mehreren aktiven Schaltfunktionen)

Für die Analyse erweist es sich als zweckmäßig, die parameterabhängigen Lösungen des hybriden Systems links und rechts von  $p^*$  getrennt zu betrachten. In der Nähe von  $p^*$  fächern die parameterabhängigen Schaltzeitpunkte  $t^i(p)$  die  $p$ - $t$ -Ebene in verschiedene Bereiche auf, so dass für  $p \neq p^*$  die Lösungen eines mehrschichtigen parameterabhängigen Systems zutage treten. Daher bezeichnen wir die Lösung für  $p < p^*$  in der  $i$ -ten Schicht mit  $x^{i-}$ . Die entsprechende Schaltfunktion wird mit  $q^{i-}$  und die Zustandsänderung mit  $A^{i-}$  benannt. Wir setzen voraus, dass die Funktionen  $q^{i-}$ ,  $A^{i-}$  in einer ganzen Umgebung von  $(x^{i-1,-}(p^*, t^*), p^*, t^*)$  existieren und dort glatt seien. Dasselbe gelte für die rechte Seite  $f^{i-}$  in einer Umgebung von  $(x^{i-}(p^*, t^*), p^*, t^*)$ .



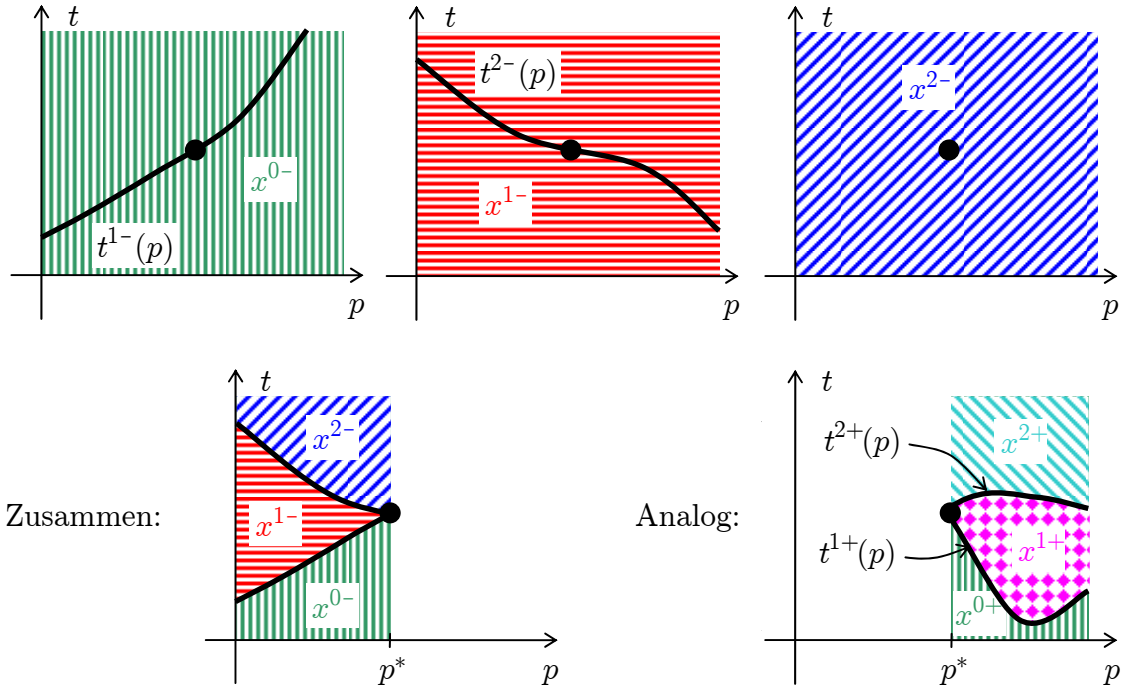


Abbildung 4.3: Schrittweiser Lösungsaufbau in der Umgebung eines Punktes  $(p^*, t^*)$  mit mehreren gleichzeitig aktiven Schaltfunktionen. Getrennte Sichtweise für  $p < p^*$  und  $p > p^*$ .

In analoger Weise seien die Größe  $x^{i+}$  sowie die Funktionen  $q^{i+}$ ,  $A^{i+}$ ,  $f^{i+}$  definiert. Es wird vorausgesetzt, dass die Regularitätsbedingungen

$$[q_x^{i\pm} f^{i-1,\pm} + q_t^{i\pm}] (x^{i-1,\pm}(p^*, t^*), p^*, t^*) \neq 0 \quad (4.7)$$

für  $i = 1, \dots, n_q$  erfüllt seien. Damit gibt es eindeutige parameterabhängige Schaltzeitpunkte  $t^{i\pm}(p)$ . Wenn wir  $n_{x^0} = n_{x^{i-1}}$  annehmen, können wir für die partiellen Ableitungen  $q_{x^{i-1},\pm}^{i\pm}$  kurz  $q_x^{i\pm}$  schreiben. Induktiv folgt aus Bemerkung 2.11 für  $i = 1, \dots, n_q$ : Es gibt eine Umgebung von  $(p^*, t^*)$ , in der Lösungen  $x^{i-}$  existieren (also auch für  $p > p^*$ ), die jeweils das System gewöhnlicher Differentialgleichungen

$$\dot{x}^{i-} = f^{i-}(x^{i-}, p, t)$$

sowie die Anfangsbedingung

$$x^{i-}(p, t^{i-}(p)) = A^{i-}(x^{i-1,-}(p, t^{i-}(p)), p, t^{i-}(p))$$

erfüllen. Aus den vereinbarten Bezeichnungen ergibt sich in einer Umgebung von  $p^*$ :

$$t^{1-}(p) \leq \dots \leq t^{n_q-}(p) \quad \text{für } p \leq p^* \quad \text{sowie} \quad t^{1+}(p) \leq \dots \leq t^{n_q+}(p) \quad \text{für } p \geq p^*.$$

Die beschriebene Situation ist in Abbildung 4.3 grafisch dargestellt, vgl. auch Abbildung 4.2. Es sei  $U$  eine hinreichend kleine Umgebung von  $(p^*, t^*)$ . Für  $i = 1, \dots, n_q - 1$

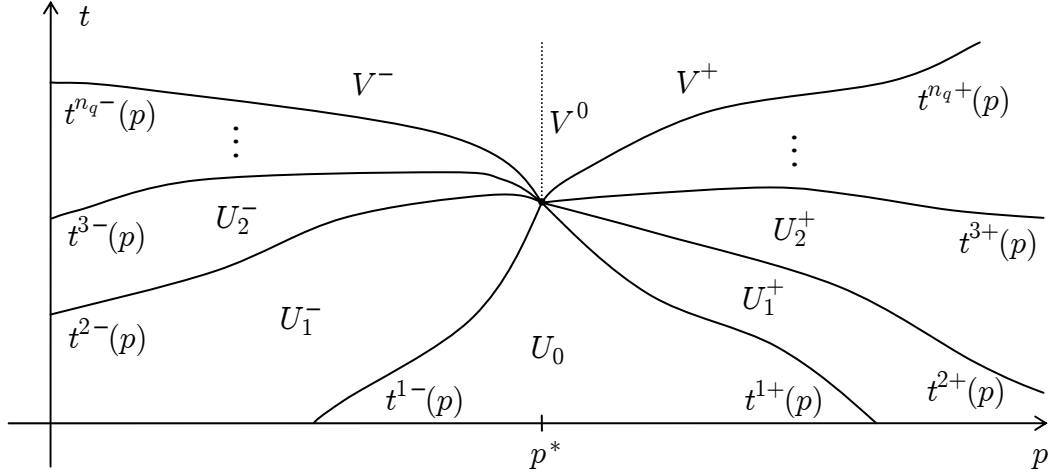


Abbildung 4.4: Aufteilung der Definitionsbereiche in der Umgebung des Ereignispunkts bei mehreren gleichzeitig aktiven Schaltfunktionen.

trägt jede Lösung  $x^{i-}$  auf der Menge  $U_i^-$  und jede Lösung  $x^{i+}$  auf der Menge  $U_i^+$  mit

$$\begin{aligned} U_i^- &:= U \cap \{(p, t) : t^{i-}(p) \leq t < t^{i+1,-}(p), p < p^*\}, \\ U_i^+ &:= U \cap \{(p, t) : t^{i+}(p) \leq t < t^{i+1,+}(p), p > p^*\} \end{aligned}$$

zur Gesamtlösung bei. Zudem liefert  $x^0$  auf

$$U_0 := U \cap \left( \{(p, t) : t < t^{1-}(p), p \leq p^*\} \cup \{(p, t) : t < t^{1+}(p), p > p^*\} \right)$$

einen Beitrag, siehe Abbildung 4.4. Es bleiben noch die Mengen

$$\begin{aligned} V^- &:= U \cap \{(p, t) : t^{n_q-}(p) \leq t, p < p^*\}, \\ V^0 &:= U \cap \{(p^*, t) : t \geq t^*\}, \\ V^+ &:= U \cap \{(p, t) : t^{n_q+}(p) \leq t, p > p^*\}, \end{aligned}$$

mit denen die Lösungen  $x^{n_q-}$  und  $x^{n_q+}$  die Gesamtlösung auf  $V := V^- \cup V^0 \cup V^+$  vervollständigen:

$$x(p, t) := \begin{cases} x^0(p, t), & \text{falls } (p, t) \in U_0, \\ x^{i-}(p, t), & \text{falls } (p, t) \in U_i^-, \\ x^{i+}(p, t), & \text{falls } (p, t) \in U_i^+, \\ x^{n_q}(p, t), & \text{falls } (p, t) \in V, \end{cases}$$

mit

$$x^{n_q}(p, t) := \begin{cases} x^{n_q-}(p, t), & \text{falls } (p, t) \in U, p \leq p^*, \\ x^{n_q+}(p, t), & \text{falls } (p, t) \in U, p > p^*. \end{cases} \quad (4.8)$$

Auf der Menge  $V^0$  ist die Lösung über  $x^{n_q-}$  definiert. Man hätte sie genauso über  $x^{n_q+}$  definieren können, da diese Wahl für spätere Überlegungen keine prinzipielle Rolle spielt.

Wir weisen darauf hin, dass eine rechte Seite  $f^{i-}$  im Allgemeinen nicht mit einer rechten Seite  $f^{j+}$  für ein bestimmtes  $j$  übereinstimmt. Beispiel 4.31 zeigt einen solchen Fall. Ebenso wenig gilt  $t^{i-}(p) = t^{j+}(p)$  oder

$$A^{i-}\left(x^{i-1,-}(p, t^{i-}(p)), p, t^{i-}(p)\right) = A^{j+}\left(x^{j-1,+}(p, t^{j+}(p)), p, t^{j+}(p)\right)$$

in einer Umgebung von  $p^*$ . □

### Bemerkung 4.25 (Deckungsgleiche Nullstellenkurven)

Besitzt ein Modelica-Modell zwei Modellteile derselben Modellklasse und werden diese mit den gleichen Steuersignalen angesteuert, so ist es möglich, dass Schaltfunktionen der beiden Modellteile zur selben Zeit aktiv werden und dass zusätzlich die Schaltzeiten der Modellteile in gleicher Weise von Parameterwerten abhängen. Dann fallen zwei Nullstellenkurven, etwa  $t^1(p)$ ,  $t^2(p)$ , die über die Schaltfunktionen definiert sind, zusammen. Diese Situation tritt auch für Modelle ein, deren Schaltzeiten unabhängig vom untersuchten Parameter sind. Bezogen auf die mathematische Struktur bedeutet dies:

Wenn für ein  $i$  die Gleichung  $t^{i-}(p) = t^{i+1,-}(p)$  lokal für  $p < p^*$  gilt, so folgt in einer hinreichend kleinen Umgebung  $U$  die Entartung  $U_i^- = \emptyset$ . In diesem Fall fassen wir die Schichten  $i$  und  $i+1$  zu einer Schicht  $\tilde{i}$  zusammen:

$$f^{\tilde{i}-} := f^{i+1,-}, \quad q^{\tilde{i}-} := q^{i-} \quad \text{und} \quad A^{\tilde{i}-}(x^{i-1,-}, p, t) := A^{i+1,-}(A^{i-}(x^{i-1,-}, p, t), p, t).$$

Diese Zusammenfassung kann solange sukzessive fortgeführt werden, bis alle Nullstellenkurven in  $U$  genau einen gemeinsamen Schnittpunkt  $(p^*, t^*)$  besitzen. Somit können wir ohne Einschränkung  $U_i^\pm \neq \emptyset$  ( $i = 1, \dots, n_q$ ) annehmen. □

### Bemerkung 4.26 (Sensitivitäten der Schichtlösungen)

Jede Lösung  $x^{i\pm}$  besitzt nach Satz 4.7 stetige Parametersensitivitäten in einer Umgebung des Punkts  $(p^*, t^*)$ . Nach Bemerkung 4.8 lassen sich die Sensitivitäten  $x_p^{i\pm}(p^*, t^*)$  mit den Definitionen  $x^{0\pm} := x^0$  für  $i = 1, \dots, n_q$  rekursiv wie folgt berechnen:

$$\begin{aligned} x_p^{i\pm}(p^*, t^*) &= [A_x^{i\pm} x_p^{i-1,\pm}(p^*, t^*) + A_p^{i\pm}](x^{i-1,\pm}(p^*, t^*), p^*, t^*) \\ &\quad + \left( [A_x^{i\pm} f^{i-1,\pm} + A_t^{i\pm}](x^{i-1,\pm}(p^*, t^*), p^*, t^*) - f^{i\pm}(x^{i\pm}(p^*, t^*), p^*, t^*) \right) t_p^{i\pm}(p^*). \end{aligned} \quad (4.9)$$

Hierbei ist

$$t_p^{i\pm}(p^*) = - \left[ \frac{q_x^{i\pm} x_p^{i-1,\pm}(p^*, t^*) + q_p^{i\pm}}{q_x^{i\pm} f^{i-1,\pm} + q_t^{i\pm}} \right] (x^{i-1,\pm}(p^*, t^*), p^*, t^*).$$

Treten am Ereignispunkt keine unstetigen Zustandsänderungen ( $A^i = \text{id}_x$ ,  $i = 1, \dots, n_q$ ) auf, so reduziert sich Gleichung (4.9) zu:

$$\begin{aligned} x_p^{i\pm}(p^*, t^*) &= x_p^{i-1,\pm}(p^*, t^*) \\ &\quad + \left( f^{i-1,\pm}(x^{i-1,\pm}(p^*, t^*), p^*, t^*) - f^{i\pm}(x^{i\pm}(p^*, t^*), p^*, t^*) \right) t_p^{i\pm}(p^*). \end{aligned} \quad (4.10)$$

□

**Satz 4.27 (Sensitivitäten bei mehreren aktiven Schaltfunktionen)**

Ein mehrschichtiges parameterabhängiges System habe zur Zeit  $t^*$  für den Parameterwert  $p^*$  mehrere aktive Schaltfunktionen, für die jeweils die Regularitätsbedingungen (4.7) erfüllt seien. Es gelte weiter

$$x^{n_q-}(p^*, t^*) = x^{n_q+}(p^*, t^*)$$

sowie  $f^{n_q} := f^{n_q-} = f^{n_q+}$  in einer Umgebung von  $(x^{n_q-}(p^*, t^*), p^*, t^*)$ . Dann ist die Lösung  $x^{n_q}$  genau dann in  $V$  stetig differenzierbar, wenn

$$x_p^{n_q-}(p^*, t^*) = x_p^{n_q+}(p^*, t^*)$$

gilt. Falls eine der äquivalenten Bedingungen erfüllt ist, so gelten die folgenden Zusammenhänge:

$$\dot{x}_p^{n_q} = f_x^{n_q}(x^{n_q}, p, t)x_p^{n_q} + f_p^{n_q}(x^{n_q}, p, t), \quad x_p^{n_q}(p^*, t^*) = x_p^{n_q-}(p^*, t^*) = x_p^{n_q+}(p^*, t^*).$$

**Beweis:**

a) Wir setzen voraus, dass  $x^{n_q}$  auf  $V$  stetig differenzierbar ist. Dann folgt aus der Definition von  $x^{n_q}$  in Gleichung (4.8) und aus der Stetigkeit von  $x_p^{n_q+}$  eine Richtung der Äquivalenzaussage des Satzes.

b) Wir setzen  $x_p^{n_q-}(p^*, t^*) = x_p^{n_q+}(p^*, t^*) =: x_p^*$  voraus und bezeichnen  $x^* := x^{n_q-}(p^*, t^*) = x^{n_q+}(p^*, t^*)$ . Aus der Eindeutigkeit der Lösung der Anfangswertprobleme

$$\begin{aligned} \dot{x}^{n_q-} &= f^{n_q-}(x^{n_q-}, p^*, t), & x^{n_q-}(p^*, t^*) &= x^*, \\ \dot{x}^{n_q+} &= f^{n_q+}(x^{n_q+}, p^*, t), & x^{n_q+}(p^*, t^*) &= x^* \end{aligned}$$

folgt zusammen mit der Gleichheit der rechten Seiten  $f^{n_q-}, f^{n_q+}$  die Identität  $x^{n_q+} = x^{n_q-}$  auf  $V^0$ . Damit ist  $x^{n_q}$  stetig auf  $V$ . Nach Bemerkung 4.26 existieren für  $x^{n_q-}$  und  $x^{n_q+}$  stetige Parametersensitivitäten auf  $U$ . Sie erfüllen insbesondere auf  $V^0$  die Sensitivitätsgleichungen, die wegen der Stetigkeit von  $x^{n_q}$  auf  $V$  wie folgt geschrieben werden können:

$$\begin{aligned} \dot{x}_p^{n_q-} &= f_x^{n_q-}(x^{n_q}(p^*, t), p^*, t)x_p^{n_q-} + f_p^{n_q-}(x^{n_q}(p^*, t), p^*, t), \\ \dot{x}_p^{n_q+} &= f_x^{n_q+}(x^{n_q}(p^*, t), p^*, t)x_p^{n_q+} + f_p^{n_q+}(x^{n_q}(p^*, t), p^*, t). \end{aligned}$$

Zusammen mit  $f^{n_q-} = f^{n_q+}$  in einer Umgebung von  $(x^*, p^*, t^*)$  folgt, dass die rechten Seiten dieser Differentialgleichungen für  $t \geq t^*$  übereinstimmen. Mit dem jeweils gleichen Anfangswert  $x_p^*$  ergibt sich analog zu oben die Gleichung  $x_p^{n_q-} = x_p^{n_q+}$  auf  $V^0$  und damit insgesamt die stetige Differenzierbarkeit von  $x^{n_q}$  auf  $V$ . ■

In unseren bisherigen Überlegungen dieses Abschnitts ist die Schaltstruktur des mehrschichtigen hybriden Systems in der Umgebung von  $(p^*, t^*)$  bekannt. Wir gehen jetzt einen Schritt zurück und knüpfen an die Systeme aus Abschnitt 3.2.2 an, deren eindeutige Lösbarkeit am Ereignispunkt durch nachprüfbare Kriterien der Problemstruktur gesichert ist. Da die Abhängigkeit bzgl. eines Parameters neben der eindeutigen Lösbarkeit einen zusätzlichen Aspekt der Systemlösung beleuchtet, ist zu erwarten, dass für die Existenz von Sensitivitäten weitere Voraussetzungen an die Struktur zu stellen sind. Hier erweisen sich die reduziert gekoppelten Systeme mit  $\ell = 0$  als adäquate Problemklasse, die die einfach gekoppelten Systeme aus Korollar 3.27 miteinschließt.

**Satz 4.28 (Sensitivitäten für reduziert gekoppelte Systeme mit  $\ell = 0$ )**

Gegeben sei ein parameterabhängiges reduziert gekoppeltes System mit  $\ell = 0$  (vgl. Bemerkung 3.26):

$$\begin{pmatrix} \dot{\xi}^0 \\ \dot{\xi}^1 \\ \vdots \\ \dot{\xi}^m \end{pmatrix} = \dot{x} = f^s(x, p, t) = \begin{pmatrix} \varphi^0(x, p, t) \\ \varphi^{1, s_1}(x, p, t) \\ \vdots \\ \varphi^{m, s_m}(x, p, t) \end{pmatrix},$$

$$q^k = q^k(\xi^0, \xi^k, p, t) \quad \text{für } k = 1, \dots, m,$$

$$q^i = q^i(x, p, t) \quad \text{für } i = m + 1, \dots, n_q.$$

Die Größen  $\xi^k$  korrespondieren zu den Schalterstellungen  $s$ , die lokal vor dem Ereigniszeitpunkt  $t^*$  gültig sind. Die Größen  $\bar{\xi}^k$  bezeichnen im Folgenden die Komponenten der Lösung, die mit den lokal für  $t > t^*$  bestimmten Schalterstellungen  $\bar{s}$  erzeugt werden. Die zugehörigen Komponenten der rechten Seite werden mit  $\phi^k$  benannt. Dann existieren die Parametersensitivitäten  $\bar{\xi}_p^k$  für  $p = p^*$ ,  $t \geq t^*$  unter folgenden Voraussetzungen:

Alle Richtungsableitungen  $D^\pm q^k$  seien in  $(x^*, p^*, t^*)$  mit  $x^* := x(p^*, t^*)$  ungleich Null, und es gebe für keine aktive Schaltfunktion den Fall 3 aus Tabelle 3.1. Die Abbildungen  $f^s$  seien für alle auftretenden Schalterstellungen glatt, ebenso die Schaltfunktionen  $q^i$  für  $i = 1, \dots, n_q$ .

Es gilt dann  $\bar{\xi}_p^0(p^*, t^*) = \xi_p^0(p^*, t^*)$  und für  $k = 1, \dots, m$ :

$$\bar{\xi}_p^k(p^*, t^*) = \xi_p^k(p^*, t^*) + [\varphi^{k, s_k} - \phi^k](x^*, p^*, t^*) t_p^k(p^*)$$

mit

$$t_p^k(p^*) = - \left[ \frac{q_{\xi^0}^k \xi_p^0(p^*, t^*) + q_{\xi^k}^k \xi_p^k(p^*, t^*) + q_p^k}{q_{\xi^0}^k \varphi^0 + q_{\xi^k}^k \varphi^{k, s_k} + q_t^k} \right] (x^*, p^*, t^*).$$

**Beweis:**

Ziel des Beweises ist, die Voraussetzungen aus Satz 4.27 zu zeigen. Dazu bringen wir das System zunächst in die Form aus Bemerkung 4.24. Für das System mit unstetiger rechter Seite verwenden wir die Bezeichnungen aus Bemerkung 3.26.

Da alle Richtungsableitungen ungleich Null sind, besitzen sie aus Stetigkeitsgründen keine Nullstellen in einer Umgebung von  $(x^*, p^*, t^*)$ . Somit ist  $\bar{s}$  unabhängig von  $p$  in einer Umgebung von  $p^*$ . Hieraus folgt insbesondere, dass  $\phi = \phi(x, p, t)$  glatt ist.

Nach Satz 4.3 existiert  $x_p^0$  zur Lösung  $x^0 := (\xi^{0\top}, \dots, \xi^{m\top})^\top$  für die Zeit  $t \leq t^*$ . Wir definieren damit

$$R_k := - \left[ \frac{q_{\xi^0}^k \xi_p^0(p^*, t^*) + q_{\xi^k}^k \xi_p^k(p^*, t^*) + q_p^k}{q_{\xi^0}^k \varphi^0 + q_{\xi^k}^k \varphi^{k, s_k} + q_t^k} \right] (x^*, p^*, t^*)$$

für  $k = 1, \dots, m$ . Ohne Einschränkung nehmen wir  $m = n_q$  an, ansonsten ignorieren wir die aktiven Schaltfunktionen  $q^i$  mit  $i = m + 1, \dots, n_q$ , da sie die Lösung lokal nicht beeinflussen. Des Weiteren seien die Komponenten  $\xi^k$  so sortiert, dass folgendes gilt:

$$R_1 \geq R_2 \geq \dots \geq R_m.$$

Wenn die  $R_k$  nicht paarweise voneinander verschieden sind, dann werden die entsprechenden Schichten gemäß Bemerkung 4.25 zusammengefasst. Man sieht sofort, dass  $q^{1-} = q^1$  gilt und  $f^{1-}$  wie in (4.11) darstellbar ist. Induktiv ergibt sich  $q^{k-} = q^k$  und  $f^{k-}$  sowie analog  $q^{k+} = q^{m-k+1}$  und  $f^{k+}$ :

$$f^{1-} := \begin{pmatrix} \phi^0 \\ \phi^1 \\ \varphi^{2,s_2} \\ \vdots \\ \vdots \\ \varphi^{m,s_m} \end{pmatrix}, \quad f^{k-} := \begin{pmatrix} \phi^0 \\ \vdots \\ \phi^k \\ \varphi^{k+1,s_{k+1}} \\ \vdots \\ \varphi^{m,s_m} \end{pmatrix}, \quad f^{k+} := \begin{pmatrix} \varphi^0 \\ \vdots \\ \varphi^{m-k,s_{m-k}} \\ \phi^{m-k+1} \\ \vdots \\ \phi^m \end{pmatrix}. \quad (4.11)$$

Da die vorausgesetzte Problemklasse keine unstetigen Zustandsänderungen enthält gilt offensichtlich  $A^{k\pm} = \text{id}_x$ . Wir berechnen die Sensitivitäten gemäß Gleichung (4.10). Wegen  $t_p^{1-}(p^*) = R_1$  folgern wir induktiv für  $k = 1, \dots, m$ :

$$\begin{aligned} x_p^{k-}(p^*, t^*) &= x_p^{k-1,-}(p^*, t^*) + [f^{k-1,-} - f^{k-}](x^*, p^*, t^*) \cdot t_p^{k-}(p^*) \\ &= x_p^{k-1,-}(p^*, t^*) + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \varphi^{k,s_k} - \phi^k \\ 0 \\ \vdots \\ 0 \end{pmatrix} (x^*, p^*, t^*) \cdot t_p^{k-}(p^*) = \begin{pmatrix} \bar{\xi}_p^{0-} \\ \vdots \\ \bar{\xi}_p^{k-} \\ \xi_p^{k+1} \\ \vdots \\ \xi_p^m \end{pmatrix} (p^*, t^*). \end{aligned}$$

Daraus folgt für  $k < m$  jeweils  $t_p^{k+1,-}(p^*) = R_{k+1}$ . Analog erhält man

$$\begin{aligned} x_p^{k+}(p^*, t^*) &= x_p^{k-1,+}(p^*, t^*) + [f^{k-1,+} - f^{k+}](x^*, p^*, t^*) \cdot t_p^{k+}(p^*) \\ &= x_p^{k-1,+}(p^*, t^*) + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \varphi^{\alpha,s_\alpha} - \phi^\alpha \\ 0 \\ \vdots \\ 0 \end{pmatrix} (x^*, p^*, t^*) \cdot t_p^{k+}(p^*) = \begin{pmatrix} \xi_p^0 \\ \vdots \\ \xi_p^{m-k} \\ \bar{\xi}_p^{\alpha,+} \\ \vdots \\ \bar{\xi}_p^{m+} \end{pmatrix} (p^*, t^*) \end{aligned}$$

mit  $\alpha := m - k + 1$  und  $t_p^{k+}(p^*) = R_{m-k+1}$ .

Die Identität  $x^{m-}(p^*, t^*) = x^{m+}(p^*, t^*)$  ist trivialerweise erfüllt, zudem folgt wegen  $\varphi^0 = \phi^0$  die Gleichung  $f^{m-} = \phi = f^{m+}$  in einer Umgebung von  $(x^*, p^*, t^*)$ . Zu guter Letzt folgt aus obigen Überlegungen an der Stelle  $(p^*, t^*)$ :

$$x_p^{m-} = \begin{pmatrix} \bar{\xi}_p^{0-} \\ \bar{\xi}_p^{1-} \\ \bar{\xi}_p \\ \vdots \\ \bar{\xi}_p^{m-} \end{pmatrix} = \begin{pmatrix} \xi_p^0 \\ \xi_p^1 \\ \vdots \\ \xi_p^m \end{pmatrix} + \begin{pmatrix} 0 \\ (\varphi^{1,s_1} - \phi^1)t_p^1(p^*) \\ \vdots \\ (\varphi^{m,s_m} - \phi^m)t_p^m(p^*) \end{pmatrix} = \begin{pmatrix} \bar{\xi}_p^{0+} \\ \bar{\xi}_p^{1+} \\ \bar{\xi}_p \\ \vdots \\ \bar{\xi}_p^{m+} \end{pmatrix} = x_p^{m+}.$$

Somit sind alle Voraussetzungen von Satz 4.27 erfüllt und daher alle Behauptungen bewiesen. Es bleibt anzumerken, dass für die Berechnung der Sensitivitäten am Ereignispunkt keine Sortierung der Komponenten  $\xi^k$  nötig ist, da alle relevanten Werte von der Reihenfolge der Teilsysteme  $1, \dots, m$  unabhängig sind. Für die praktische Berechnung muss folglich auch keine Zusammenfassung gemäß Bemerkung 4.25 durchgeführt werden. ■

### Bemerkung 4.29 (Sensitivitäten mit aktiven regulären Schaltfunktionen)

Der Fall genau einer aktiven *regulären* Schaltfunktion wurde bereits in Abschnitt 4.1.4 besprochen. Die Argumentation im Beweis von Satz 4.28 lässt sich auch für reguläre Schaltfunktionen führen, wenn man wie in Bemerkung 3.29 als aktive Schaltfunktion einen beliebigen, aber festen glatten Ast der regulären Schaltfunktion verwendet. □

Die Voraussetzungen in Satz 4.28 sind so gewählt, dass jede Komponente der rechten Seite des unstetigen Systems höchstens von einem aktiven Schalter abhängt. Damit ist für jede Sensitivitätskomponente eine eindeutige Zuordnung zu einer Ableitung  $t_p^k(p^*)$  garantiert. Beispiel 4.31 wird im nächsten Abschnitt demonstrieren, dass die Sensitivitäten im Allgemeinen nicht existieren, wenn Zustandsableitungen von mehreren aktiven Schaltern abhängen.

## 4.2.2 Beispiele

### Beispiel 4.30 (Entkoppelte Systeme)

Ein natürlicher Test, die Aussagen von Satz 4.28 zu prüfen, ist die Zusammenfassung zweier völlig entkoppelter Systeme, die zufällig zum selben Zeitpunkt aktive Schaltfunktionen besitzen. Es seien also die folgenden Systeme gegeben:

$$\begin{aligned}\dot{\xi}^1 &= \varphi^{1,s_1}(\xi^1, p, t) = \begin{cases} \varphi^{1,+1}(\xi^1, p, t), & \text{falls } q^1(\xi^1, p, t) > 0, \\ \varphi^{1,-1}(\xi^1, p, t), & \text{falls } q^1(\xi^1, p, t) < 0, \end{cases} \\ \dot{\xi}^2 &= \varphi^{2,s_2}(\xi^2, p, t) = \begin{cases} \varphi^{2,+1}(\xi^2, p, t), & \text{falls } q^2(\xi^2, p, t) > 0, \\ \varphi^{2,-1}(\xi^2, p, t), & \text{falls } q^2(\xi^2, p, t) < 0. \end{cases}\end{aligned}$$

Es gelte:  $q^1(\xi^1(p^*, t^*), p^*, t^*) = 0$ ,  $q^2(\xi^2(p^*, t^*), p^*, t^*) = 0$ . Bei getrennter Betrachtung der Systeme errechnen sich die Sensitivitäten am Ereignispunkt nach Satz 4.7 und Bemerkung 4.8 mit den rechten Seiten  $\phi^k$  für  $t \geq t^*$  und  $k = 1, 2$  wie folgt:

$$\bar{\xi}_p^k(p^*, t^*) = \xi_p^k(p^*, t^*) + [\varphi^{k,s_k} - \phi^k](\xi^k(p^*, t^*), p^*, t^*) t_p^k(p^*).$$

Dabei gilt

$$t_p^k(p^*) = - \left[ \frac{q_{\xi^k}^k \xi_p^k(p^*, t^*) + q_p^k}{q_{\xi^k}^k \varphi^{k,s_k} + q_t^k} \right] (\xi^k(p^*, t^*), p^*, t^*).$$

Dies entspricht genau den Zusammenhängen, die sich aus Satz 4.28 mit  $n_{\xi^0} = 0$  ergeben, wenn die beiden Systeme in einem System geschrieben werden:

$$\dot{x} = \begin{pmatrix} \dot{\xi}^1 \\ \dot{\xi}^2 \end{pmatrix} = \begin{pmatrix} \varphi^{1,s_1}(\xi^1, p, t) \\ \varphi^{2,s_2}(\xi^2, p, t) \end{pmatrix} = f^s(x, p, t).$$

□

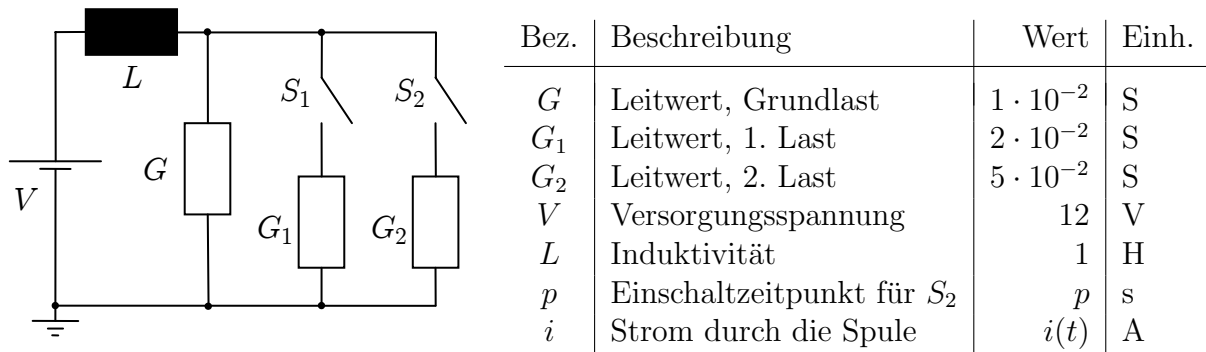


Abbildung 4.5: Schaltkreis für paralleles Einschalten von Verbrauchern mit Tabelle der verwendeten Größen und Parameter, siehe Beispiel 4.31.

### Beispiel 4.31 (Gleichzeitiges Zuschalten von elektrischen Verbrauchern)

Nachdem im vorangegangenen Beispiel gezeigt wurde, wie Satz 4.28 angewendet werden kann, betrachten wir nun ein Modell, das die Voraussetzungen des Satzes nicht erfüllt. Die elektrische Schaltung aus Abbildung 4.5 beschreibt zuschaltbare Lasten  $G_1$  und  $G_2$ , die über die Schalter  $S_1$ ,  $S_2$  parallel zu einer Grundlast  $G$  an eine Gleichspannungsquelle und eine lineare Induktivität  $L$  angeschlossen sind. Alle Lasten sind durch ihre linearen Leitwerte bekannt. Zunächst seien beide Schalter offen. Schalter  $S_1$  schließt nach einer halben Sekunde und erhöht die Gesamtlast. Der Zeitpunkt, wann Schalter  $S_2$  die Last  $G_2$  hinzuschaltet, sei variabel und über den Parameter  $p$  festgelegt. Der Zeitverlauf des Stroms  $i$  durch die Spule wird über die lineare Differentialgleichung

$$\frac{di}{dt} = \frac{1}{L} \left( V - \frac{1}{G + G_1^* + G_2^*} i \right)$$

mit

$$G_1^* := \begin{cases} 0, & \text{falls } t < \frac{1}{2}, \\ G_1, & \text{falls } t > \frac{1}{2}, \end{cases} \quad G_2^* := \begin{cases} 0, & \text{falls } t < p, \\ G_2, & \text{falls } t > p \end{cases}$$

und dem Anfangswert  $i(0) = 0$  modelliert. Das System hat eine unstetige rechte Seite mit mehreren Schaltern nach Definition 3.12. Für  $p^* = \frac{1}{2}$  sind die beiden Schaltfunktionen  $t - \frac{1}{2}$  und  $t - p$  zur Zeit  $t^* = \frac{1}{2}$  gleichzeitig aktiv. Da beide Schalter die Zeitableitung von  $i$  beeinflussen, gehört dieses Modell nicht zur Problemklasse aus Satz 4.28. Die Lösung des Problems lässt sich elementar berechnen und lautet in Abhängigkeit von  $p$  und  $t$ :

$0 < p \leq \frac{1}{2}$	$\frac{1}{2} < p$
$i(p, t) = \begin{cases} i^0(p, t), & \text{falls } 0 \leq t < p, \\ i^{1-}(p, t), & \text{falls } p \leq t < \frac{1}{2}, \\ i^{2-}(p, t), & \text{falls } \frac{1}{2} \leq t, \end{cases}$	$i(p, t) = \begin{cases} i^0(p, t), & \text{falls } 0 \leq t < \frac{1}{2}, \\ i^{1+}(p, t), & \text{falls } \frac{1}{2} \leq t < p, \\ i^{2+}(p, t), & \text{falls } p \leq t \end{cases}$



mit

$$\begin{aligned}
i^0(p, t) &= VG \left( 1 - e^{-\frac{t}{LG}} \right), \\
i^{1-}(p, t) &= V \left( G + G_2 - \left( G_2 + Ge^{-\frac{p}{LG}} \right) e^{-\frac{t-p}{L(G+G_2)}} \right), \\
i^{2-}(p, t) &= V \left( G + G_1 + G_2 - \left( G_1 + \left( G_2 + Ge^{-\frac{p}{LG}} \right) e^{-\frac{\frac{1}{2}-p}{L(G+G_2)}} \right) e^{-\frac{t-\frac{1}{2}}{L(G+G_1+G_2)}} \right), \\
i^{1+}(p, t) &= V \left( G + G_1 - \left( G_1 + Ge^{-\frac{1}{2LG}} \right) e^{-\frac{t-\frac{1}{2}}{L(G+G_1)}} \right), \\
i^{2+}(p, t) &= V \left( G + G_1 + G_2 - \left( G_2 + \left( G_1 + Ge^{-\frac{1}{2LG}} \right) e^{-\frac{p-\frac{1}{2}}{L(G+G_1)}} \right) e^{-\frac{t-p}{L(G+G_1+G_2)}} \right).
\end{aligned}$$

Hierbei werden die Schreibweisen aus Abschnitt 4.2.1 verwendet. Aus den Lösungsästen errechnen sich die Sensitivitäten wie folgt:

$$\begin{aligned}
i_p^0(p, t) &= 0, \\
i_p^{1-}(p, t) &= \frac{VG_2}{L(G+G_2)} \left( e^{-\frac{p}{LG}} - 1 \right) e^{-\frac{t-p}{L(G+G_2)}}, \\
i_p^{2-}(p, t) &= \frac{VG_2}{L(G+G_2)} \left( e^{-\frac{p}{LG}} - 1 \right) e^{-\frac{\frac{1}{2}-p}{L(G+G_2)} - \frac{t-\frac{1}{2}}{L(G+G_1+G_2)}}, \\
i_p^{1+}(p, t) &= 0, \\
i_p^{2+}(p, t) &= \frac{VG_2}{L(G+G_1+G_2)} \left( \frac{1}{G+G_1} \left( G_1 + Ge^{-\frac{1}{2LG}} \right) e^{-\frac{p-\frac{1}{2}}{L(G+G_1)}} - 1 \right) e^{-\frac{t-p}{L(G+G_1+G_2)}}.
\end{aligned}$$

Der Lösungsverlauf und die Sensitivitäten sind in Abbildung 4.6 zusammen mit den parameterabhängigen Kurven für die Schaltzeitpunkte  $t(p) = \frac{1}{2}$  und  $t(p) = p$  in einer Umgebung um den Punkt  $(p^*, t^*)$  dargestellt. Der farbcodierte Verlauf der Sensitivitäten zeigt, dass die Lösung auf  $\{(p^*, t) : t > t^*\}$  nicht differenzierbar nach  $p$  ist. Rechts- und linksseitige Grenzwerte der Differenzenquotienten stimmen nicht überein. Insbesondere gelten:

$$\begin{aligned}
i_p^{2-}(p^*, t^*) &= \frac{VG_2}{L(G+G_2)} \left( e^{-\frac{1}{2LG}} - 1 \right) = -10.0, \\
i_p^{2+}(p^*, t^*) &= \frac{VG_2}{L(G+G_1+G_2)(G+G_1)} \left( e^{-\frac{1}{2LG}} - 1 \right) = -2.5.
\end{aligned}$$

Will man diese Sensitivitäten mit Hilfe von Bemerkung 4.26 berechnen, so benötigt man die folgenden Funktionen, die die Werte in den letzten beiden Zeilen der Tabelle zur Folge haben:

$k$	0	1−	2−	1+	2+
$q^k$	—	$t - p$	$t - \frac{1}{2}$	$t - \frac{1}{2}$	$t - p$
$f^k$	$\frac{1}{L} \left( V - \frac{1}{G} i \right)$	$\frac{1}{L} \left( V - \frac{1}{G+G_2} i \right)$	$\frac{1}{L} \left( V - \frac{1}{G+G_1+G_2} i \right)$	$\frac{1}{L} \left( V - \frac{1}{G+G_1} i \right)$	$f^{2-}$
$A^k$	0	$\text{id}_i$	$\text{id}_i$	$\text{id}_i$	$\text{id}_i$
$t_p^k(p^*)$	—	1	0	0	1
$i_p^k(p^*, t^*)$	0	−10.0	−10.0	0	−2.5

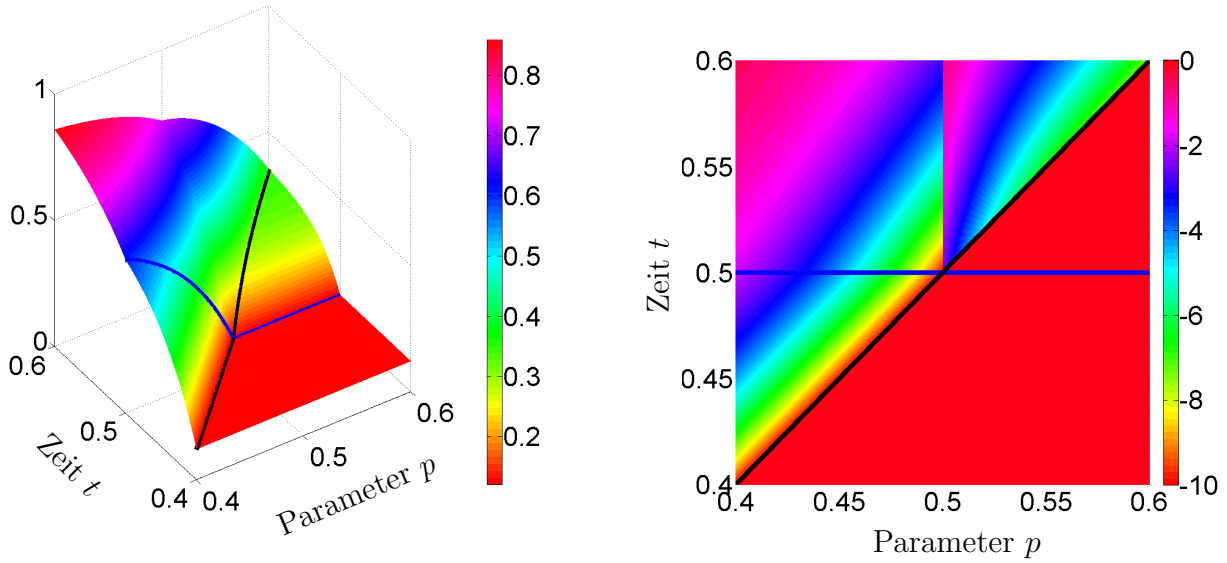


Abbildung 4.6: Lösungsverlauf  $i(p, t)$  (links) und Parameterableitung  $i_p(p, t)$  (rechts) für Beispiel 4.31. Blaue Kurve: Schaltfläche zu  $t(p) = \frac{1}{2}$ . Schwarze Kurve: Schaltfläche zu  $t(p) = p$ . Die Schaltflächen sind in den Lösungsflächen eingetragen.

Sowohl die Berechnung über die analytische Lösung als auch die Anwendung von Satz 4.27 führen wegen  $i_p^{2-}(p^*, t^*) \neq i_p^{2+}(p^*, t^*)$  zu dem Ergebnis, dass der Strom  $i$  für die Zeit nach dem zeitgleichen Umschalten keine Parametersensitivitäten besitzt.  $\square$

### Beispiel 4.32 (Nichtglatte Verläufe der Schaltzeitpunkte)

Das Ziel dieses Beispiels ist zu zeigen, dass die Verläufe der Schaltzeitpunkte in Abhängigkeit eines Parameters an mehrfachen Ereignispunkten nichtglatt aneinander schließen können. Dazu ist ein eindimensionales System mit zustandsabhängiger Schaltfunktion ausreichend:

$$\dot{x} = a + b, \quad x(0) = 0, \quad p^* = 2, \quad t^* = 3$$

mit

$$a = \begin{cases} 9, & \text{falls } t < 39 - x, \\ p, & \text{falls } t > 39 - x \end{cases} \quad \text{und} \quad b = \begin{cases} p + 1, & \text{falls } t < p + 1, \\ 1, & \text{falls } t > p + 1. \end{cases}$$

Die analytische Lösung errechnet sich für  $0 < p \leq 2$  wie folgt:

$$x(p, t) = \begin{cases} (p + 10)t, & \text{falls } t \leq p + 1, \\ 10t + p^2 + p, & \text{falls } p + 1 \leq t \leq \frac{-p^2 - p + 39}{11}, \\ (p + 1)t + \frac{1}{11}p^3 + \frac{3}{11}p^2 - \frac{37}{11}p + \frac{351}{11}, & \text{falls } \frac{-p^2 - p + 39}{11} \leq t. \end{cases}$$

Im Fall  $p > 2$  gilt:

$$x(p, t) = \begin{cases} (p + 10)t, & \text{falls } t \leq \frac{39}{p+11}, \\ (2p + 1)t + \frac{351 - 39p}{p+11}, & \text{falls } \frac{39}{p+11} \leq t \leq p + 1, \\ (p + 1)t + p^2 + p + \frac{351 - 39p}{p+11}, & \text{falls } p + 1 \leq t. \end{cases}$$

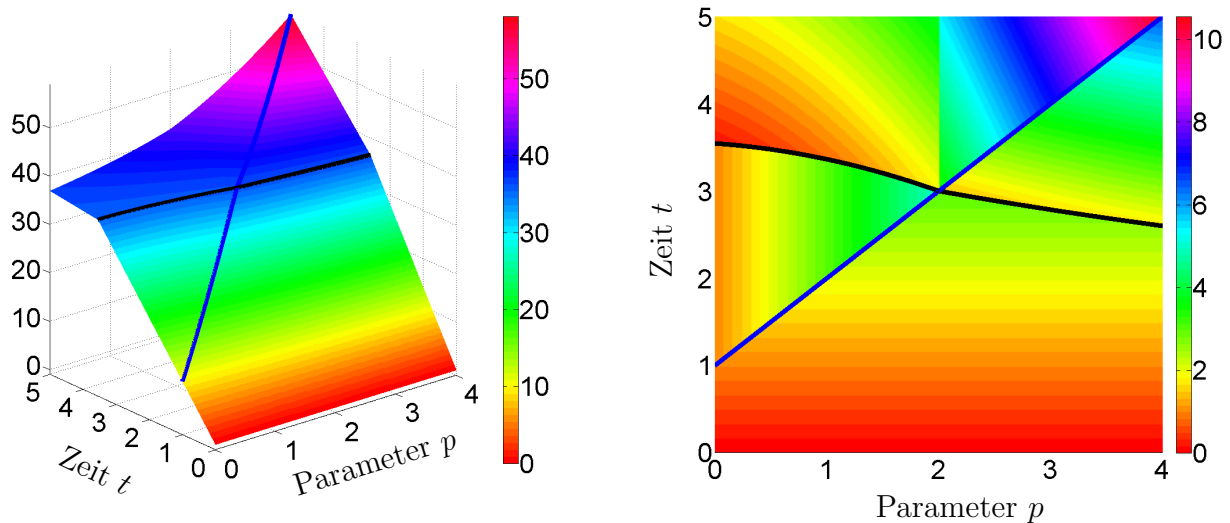


Abbildung 4.7: Lösungsverlauf  $x(p, t)$  (links) und Parameterableitung  $x_p(p, t)$  (rechts) für Beispiel 4.32. Blaue Kurve: Schaltfläche zu  $q(p, t) = t - p - 1$ . Schwarze Kurve: Schaltfläche zu  $q(x, t) = t + x - 39$  ist im Ereignispunkt  $(p^*, t^*) = (2, 3)$  nicht differenzierbar.

Untersucht man das Beispiel mit den Methoden aus Abschnitt 4.2.1, so erhält man folgende Funktionen und Werte:

$k$	0	1−	2−	1+	2+
$q^k$	—	$t - p - 1$	$t + x - 39$	$t + x - 39$	$t - p - 1$
$f^k$	$p + 10$	10	$p + 1$	$2p + 1$	$p + 1$
$A^k$	0	$\text{id}_x$	$\text{id}_x$	$\text{id}_x$	$\text{id}_x$
$t_p^k(p^*)$	—	1	$-\frac{5}{11}$	$-\frac{3}{13}$	1
$x_p^k(p^*, t^*)$	3	5	$\frac{20}{11}$	$\frac{18}{13}$	$\frac{44}{13}$

Wie in Beispiel 4.31 gilt  $x_p^{2-}(p^*, t^*) \neq x_p^{2+}(p^*, t^*)$ , somit ist die Lösung lokal für  $t > t^*$  nicht nach  $p$  differenzierbar (Satz 4.27). Zusätzlich fällt hier  $t_p^{2-}(p^*) \neq t_p^{1+}(p^*)$  auf. Alle Eigenschaften des Problems lassen sich auch in Abbildung 4.7 an der Lösung bzw. Parameterableitung mit den parameterabhängigen Schaltzeitpunkten  $t(p) = p + 1$  und  $t(p) = 39 - x(p, t(p))$  erkennen. Selbst bei diesem einfachen Beispiel mit *glatten* Schaltfunktionen und abschnittsweise konstanten rechten Seiten erhält man Kurven  $t(p)$ , die am Ereignispunkt jeweils nicht differenzierbar ineinander übergehen.  $\square$

Abschnitt 4.2 zeigt, dass Zusatzvoraussetzungen an unstetige Systeme mit mehreren aktiven Schaltfunktionen zu stellen sind, um die Existenz von Parametersensitivitäten zu garantieren. Selbst für den „Standard“-Fall *einer* aktiven Schaltfunktion ergeben sich in Abschnitt 4.1.2 unterschiedliche Aspekte, je nachdem, ob die glatten Teillösungen separat oder als zusammengesetzte Lösung betrachtet werden. Die Ergebnisse der theoretischen Untersuchungen in Kapitel 4 bilden die Grundlage für die im nachfolgenden Kapitel betrachteten Verfahren zur numerischen Berechnung der Sensitivitäten unstetiger Modelle.

## 5 Numerische Sensitivitätsanalyse unstetiger Modelica-Modelle

Die Numerik gewöhnlicher Differentialgleichungen und differential-algebraischer Gleichungen ist eine in den letzten drei Jahrzehnten tief erforschte Thematik (siehe etwa [HW96, KM06] und die darin enthaltenen Referenzen). Da die virtuelle, auf Software basierende Untersuchung in der Entwicklung neuer Produkte und Verfahren eine entscheidende Rolle spielt, wächst die Anzahl und die Vielfalt von Anwendungen, die auf den numerischen Verfahren zur Integration der dynamischen Systeme aufbauen. Praktisch alle anspruchsvollen Implementierungen von Zeitintegrationsverfahren enthalten Algorithmen mit lösungsangepasstem, variablem Zeitgitter und stetiger Lösungsdarstellung. Des Weiteren unterstützen Routinen die Schaltpunktsuche für vorgegebene Schaltfunktionen, die mit ihren Nullstellen Ereignisse an Unstetigkeitsstellen anzeigen [ST00]. Nach einem Ereignispunkt beginnt die Integration wie am Anfangszeitpunkt von Neuem. Die genannten Techniken sind etwa im Programmpaket DASSL [Pet82a] implementiert, dem das BDF-Verfahren zu Grunde liegt. Dieser Code und weitere Pakete (LSODAR aus ODEPACK [Hin83] und Runge-Kutta-Verfahren aus GODESS [Ols98]) sind in der Software Dymola [Dym] eingebunden und werden dazu verwendet, Modelica-Modelle numerisch zu integrieren.

Im genannten Umfeld wollen wir numerische Software entwickeln, die sowohl Filippov-Lösungen als auch die Sensitivitäten der in Kapitel 3 konstruierten Lösungen der hybriden Systeme berechnet. Im ersten Abschnitt stellen wir grundsätzliche Möglichkeiten vor, Sensitivitäten numerisch zu berechnen. Wir gehen dabei auf die Integration der Sensitivitätsgleichungen mit dem Programmpaket DASPK3.1 [LP99] näher ein, da wir diesen Ansatz für die numerischen Untersuchungen in der Arbeit wählen. Die Einbettung von DASPK3.1 in die neu entwickelte Softwareumgebung FISEMO sowie die Beschreibung der weiteren Bestandteile sind Gegenstand des zweiten Abschnitts. Es wird dargelegt, welche Problemklassen von der Umgebung verarbeitet werden können, bevor im letzten Teil numerische Versuche an Modelica-Modellen die Funktionalität der Implementierung zeigen und die Genauigkeit der Ergebnisse beleuchten. Solange keine Verwechslungsgefahr besteht, unterscheiden wir in den Bezeichnungen die numerische und analytische Lösung nicht.

### 5.1 Ansätze zur numerischen Sensitivitätsberechnung

Die unterschiedlichen Methodiken, die Sensitivitäten  $x_p$  numerisch zu berechnen, beschreiben wir für das folgende parameterabhängige Anfangswertproblem mit der generischen rechten Seite  $f : \mathbb{R}^{n_x+n_p+1} \rightarrow \mathbb{R}^{n_x}$  und den Anfangswerten  $A : \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n_x}$ :

$$\dot{x} = f(x, p, t), \quad x(t^0) = A(p). \quad (5.1)$$

### 5.1.1 Externe und interne numerische Differentiation, adjungierte Variablen

Der Begriff *externe numerische Differentiation* wurde von Bock [Boc81] geprägt und bedeutet die (näherungsweise) Differentiation der gesamten implementierten Routine eines Integrationsverfahrens. Hierfür stehen prinzipiell die beiden Möglichkeiten der finiten Differenzen oder des automatischen Differenzierens zur Verfügung. Einfachster Vertreter der finiten Differenzen ist der einseitige Differenzenquotient

$$\frac{x(p + \Delta p_j e_j, t) - x(p, t)}{\Delta p_j}, \quad j = 1, \dots, n_p, \quad \Delta p_j \neq 0,$$

der  $x_{p_j}(p, t)$  annähert. Die Vorteile dieses Ansatzes liegen klar darin, dass er sehr leicht zu implementieren und auch leicht zu parallelisieren ist. Die Parallelisierung kann bei entsprechender Verfügbarkeit von Rechenknoten die Rechenzeit stark reduzieren. Damit wird die Tatsache, dass die Methodik mit einem hohen Rechenaufwand verbunden ist, abgemildert. Der Aufwand steigt insbesondere dann an, wenn adäquate Genauigkeiten für die Sensitivitäten erreicht werden sollen. Eine Fehleranalyse des einseitigen Differenzenquotienten ergibt [HNW93] einen Fehler der Größenordnung

$$\mathcal{O}(\Delta p_j) + \mathcal{O}\left(\frac{\text{TOL}}{\Delta p_j}\right),$$

wenn TOL die vorgegebene Fehlertoleranz für den Integrationsalgorithmus bezeichnet. Die Integrationsalgorithmen mit ihren adaptiven Komponenten (Schrittweitensteuerung, Ordnungssteuerung, Newton-Iteration, Schaltpunktsuche) beschreiben im Allgemeinen keine differenzierbaren Abbildungen, so dass der obige Fehleranteil  $\mathcal{O}(\text{TOL}/\Delta p_j)$  von der unabhängigen Berechnung der numerischen Lösungen mit den unterschiedlichen Parameterwerten herrührt. Mit der Wahl  $\Delta p_j = \mathcal{O}(\sqrt{\text{TOL}})$  wird der Fehler näherungsweise minimiert, ist dann aber immer noch  $\mathcal{O}(\sqrt{\text{TOL}})$ . Folglich muss TOL für moderate Genauigkeiten der Ableitungen sehr klein gewählt werden, was zu dem beträchtlichen Anstieg im Rechenaufwand führen kann. Man stellt in der Praxis häufig fest, dass die Sensitivitäten ungenauer als die Fehlerschätzung  $\mathcal{O}(\sqrt{\text{TOL}})$  sind – insbesondere wenn Modellgleichungen und Parameter ungünstig skaliert sind.

In [SBS98] wird darauf hingewiesen, dass die externe numerische Differentiation mit automatischem Differenzieren ebenfalls für die Sensitivitäten ungenaue Ergebnisse liefert, die von adaptiven Programmbausteinen verursacht werden. Um die Differentiation der adaptiven Elemente zu vermeiden, entwickelte Bock [Boc81] das Konzept der *internen numerischen Differentiation*, das nur die Differentiation der Diskretisierungsvorschrift vorsieht. Damit erhält man die Sensitivitäten der numerischen Lösung auch für grobe Fehlertoleranzen mit einer akzeptablen Genauigkeit. Die Berechnung der Ableitungen muss hierfür mit dem Integrationsalgorithmus so verschmolzen werden, dass sich die nichtglatten Algorithmenbestandteile in jedem Integrationsschritt für die Lösung und die Sensitivitäten jeweils gleich verhalten. Deshalb bietet der Ansatz Potential für eine effiziente Implementierung und den daraus folgenden Rechenzeitgewinn im Vergleich zur externen numerischen Differentiation [Boc87, FK00].

Man kann zwischen drei Varianten der internen numerischen Differentiation unterscheiden. Im ersten Fall werden Sensitivitäten in jedem Integrationsschritt durch finite Differenzen von Lösungsvektoren zu unterschiedlichen Parameterwerten berechnet. Dieses Vorgehen wird auch als Integration mit eingefrorener Schrittweite bezeichnet [Kie98]. Für lineare Integrationsvorschriften ist dazu der zweite Ansatz äquivalent [Boc87], für den die entsprechenden finiten Differenzen auf die rechte Seite  $f$  übertragen und zusätzlich zum Nominalsystem integriert werden. Im Fall des einseitigen Differenzenquotienten erhält man die rechten Seiten

$$\frac{f(x + x_p \Delta p_j e_j, p + \Delta p_j e_j, t) - f(x, p, t)}{\Delta p_j}, \quad j = 1, \dots, n_p, \quad \Delta p_j \neq 0.$$

Der Fehler für die Sensitivitäten  $x_{p_j}$  lässt sich dadurch mit der Maschinengenauigkeit  $\text{eps}$  zu

$$\mathcal{O}(\text{TOL}) + \mathcal{O}(\Delta p_j) + \mathcal{O}\left(\frac{\text{eps}}{\Delta p_j}\right)$$

angeben. Mit  $\Delta p_j = \sqrt{\text{eps}}$  bleibt der Fehler für  $\text{TOL} \geq \sqrt{\text{eps}}$  durch  $\mathcal{O}(\text{TOL})$  beschränkt, was damit gleichbedeutend ist, dass die Genauigkeiten der Nominallösung und der Sensitivitäten gleich sind.

Aus dem Grenzübergang der finiten Differenzen resultieren in der dritten Variante die Sensitivitätsgleichungen  $\dot{x}_p = f_x x_p + f_p$ , die gleichzeitig mit den Nominalgleichungen vom Integrationsalgorithmus numerisch gelöst werden. Die Integration der Sensitivitätsgleichungen wird im nächsten Abschnitt ausführlich am Beispiel des Integrators DASK3.1 besprochen. Generell lässt sich feststellen, dass zahlreiche Implementierungen das Prinzip der internen numerischen Differentiation mehr oder weniger streng berücksichtigen. In [BBB<sup>+</sup>01] findet man eine Auflistung effizienter Umsetzungen der Sensitivitätsberechnungen.

Wenn im Rahmen einer Sensitivitätsanalyse statt der zeitabhängigen Ableitungen  $x_p(p, t)$  der Modellzustände nur die Ableitungen von Größen  $C(p) = c(x(p, t^\infty), p, t^\infty)$  benötigt werden, dann lassen sich diese Ableitungen mit Hilfe der *adjungierten Variablen*  $\lambda$  berechnen. In Abschnitt 4.1.3 werden die definierenden Gleichungen samt den Übergangsbedingungen an Ereignispunkten hybrider Systeme hergeleitet. Nun beleuchten wir die numerischen Belange der Berechnung von  $C_p$  für glatte Systeme. Zunächst nehmen wir  $C(p) \in \mathbb{R}$  an und erhalten:

$$C_p(p) = c_p(x(p, t^\infty), p, t^\infty) + \int_{t^0}^{t^\infty} \lambda^\top(t) f_p(x(p, t), p, t) dt + \lambda^\top(t^0) A_p(p) \quad (5.2)$$

mit den adjungierten Gleichungen  $\dot{\lambda} = -f_x^\top \lambda$  und  $\lambda(t^\infty) = c_x^\top(x(p, t^\infty), p, t^\infty)$ . Da der Endwert  $\lambda(t^\infty)$  vorgegeben wird, scheidet eine gleichzeitige Integration des Nominalsystems und der adjungierten Gleichungen aus. Die Lösung erfolgt daher zunächst in Vorwärtsrichtung von  $t^0$  bis  $t^\infty$  für das Nominalsystem (5.1) und im Anschluss für die adjungierten Gleichungen rückwärts von  $t^\infty$  bis  $t^0$ .

Implementierungen [LP02, SH03, Wal03] der Rückwärtsintegration setzen oft Checkpointing-Techniken ein, um den Speicherbedarf zu begrenzen, der dadurch entsteht, dass die

numerische Lösung  $x(p, t)$  für das gesamte Zeitintervall per Interpolation verfügbar sein muss. Dazu wird das Intervall in mehrere Teilintervalle aufgeteilt und innerhalb dieser die Lösung im Speicher gehalten. Gleichzeitig zur Rückwärtsintegration kann das Integral in Gleichung (5.2) durch zusätzliche Komponenten im Zustandsvektor berechnet werden. Alternativ verwendet man nach der Integration eine Quadraturformel. Nach [Boc87] sollte die numerische Berechnung der adjungierten Variablen ebenfalls das Prinzip der internen numerischen Differentiation erfüllen.

Der Berechnung von  $\lambda^\top f_p$  bzw.  $f_x^\top \lambda$  kommt besondere Bedeutung zu, falls  $f_p$  und  $f_x$  nicht analytisch vorliegen. Numerische Approximationen mit Richtungsableitungen kommen hierfür nicht in Frage, da die Ableitungsmatrizen von links mit  $\lambda^\top$  multipliziert werden. Die Diskretisierung mit finiten Differenzen erfordert daher die direkte und oft rechenintensive Approximation der vollen Matrizen  $f_x$  und  $f_p$  für jede Auswertung der rechten Seite der Rückwärtsintegration [CLP02]. Um diesen Aufwand zu verringern, sollte – falls möglich – automatisches Differenzieren verwendet werden, um die Terme  $\lambda^\top f_p$  und  $f_x^\top \lambda$  auszuwerten. Im Rückwärtsmodus des automatischen Differenzierens können diese Ausdrücke sehr effizient berechnet werden [Gri00].

Der Vorteil der adjungierten Berechnung der Ableitungen  $C_p$  liegt darin, dass der Aufwand im Wesentlichen von den Dimensionen  $n_x$  und  $n_C$  und kaum von  $n_p$  abhängt. Im Fall  $n_C > 1$  kommt für jede Komponente von  $C$  ein neuer Satz von adjungierten Variablen zum System hinzu, woraus die Gesamtdimension  $n_x n_C$  für die Rückwärtsintegration folgt, falls eine Quadraturformel benutzt wird. Daher bringt der adjungierte Ansatz Vorteile, wenn die Ableitungen von relativ wenigen Funktionen  $C_i$  bzgl. vieler Parameter gefragt sind. Die Zeitverläufe der adjungierten Variablen geben allerdings keine direkten Informationen über das Systemverhalten preis, insbesondere kann  $\frac{d}{dp}c(x(p, t), p, t)$  für  $t < t^\infty$  damit nicht berechnet werden.

### 5.1.2 Integration der Sensitivitätsgleichungen mit DASPK3.1

Wir stellen die wesentlichen Techniken vor, die im Softwarepaket DASPK3.1 [LP99] enthalten und für die vorliegende Arbeit relevant sind. Dieses Paket ist eine Weiterentwicklung des weitverbreiteten DASSL-Codes [Pet82a] zur numerischen Lösung von differentialalgebraischen Anfangswertproblemen

$$F(\dot{x}, x, t) = 0, \quad x(t^0) = A$$

mit Differentiationsindex Eins oder Null in unstrukturierter Form sowie mit Index Zwei in Hessenbergform. Das eingesetzte BDF-Verfahren [HNW93] nutzt rückwärtsgerichtete Differenzenquotienten zur Approximation der Ableitung  $\dot{x}$ . Wenn  $k > 0$  die Ordnung des Verfahrens ist, dann wird die Lösung im BDF-Ansatz an den  $k + 1$  Zeitpunkten  $\tau_{n-k+1}, \dots, \tau_{n+1}$  polynomial interpoliert. Aus der Forderung, dass das Interpolationspolynom die Differentialgleichung zur Zeit  $\tau_{n+1}$  erfüllen soll, ergibt sich die BDF-Diskretisierung für konstante Zeitschrittweiten  $h$  und  $x_n := x(\tau_n)$ :

$$F\left(\frac{1}{h} \sum_{j=1}^k \frac{1}{j} \nabla^j x_{n+1}, x_{n+1}, \tau_{n+1}\right) = 0, \quad \nabla^0 x_n := x_n, \quad \nabla^j x_{n+1} := \nabla^{j-1} x_{n+1} - \nabla^{j-1} x_n.$$



Allgemein erhält man für variable Schrittweiten  $h_n = \tau_{n+1} - \tau_n$  die Form

$$F\left(\sum_{j=0}^k \alpha_{n,k-j} x_{n+1-j}, x_{n+1}, \tau_{n+1}\right) = 0. \quad (*)$$

Die Koeffizienten  $\alpha_{n,j}$  hängen dabei von den (vorherigen) Schrittweiten  $h_{n-k+1}, \dots, h_n$  ab. Gleichung  $(*)$  muss in jedem Integrationsschritt  $n \rightarrow n+1$  nach  $x_{n+1}$  aufgelöst werden. Das definierende nichtlineare Gleichungssystem schreibt man am besten in der kompakten Form

$$F(\alpha x + \beta, x, \tau_{n+1}) = 0 \quad (5.3)$$

mit dem unbekannten Vektor  $x$ , einem skalaren Wert  $\alpha$  und einem Vektor  $\beta$  (der von den Ergebnissen  $x_{n+1-j}$  aus zurückliegenden Zeitschritten abhängt):

$$\alpha = \frac{1}{h_n} \sum_{j=1}^k \frac{1}{j}, \quad \beta = \sum_{j=1}^k \alpha_{n,k-j} x_{n+1-j}.$$

Das gedämpfte Newton-Verfahren zur Lösung der impliziten Gleichung (5.3) liest sich mit dem Iterationszähler  $m \geq 0$  wie folgt:

$$x^{(m+1)} = x^{(m)} - \gamma J^{-1}(x^{(m)}) F(\alpha x^{(m)} + \beta, x^{(m)}, \tau_{n+1}).$$

Als Startwert für die Korrekturiteration wird das Polynom, das die zurückliegenden Lösungen  $x_{n-k}, \dots, x_n$  interpoliert, an der Stelle  $\tau_{n+1}$  ausgewertet (Prädiktorwert). Aus Effizienzgründen wird die Jacobimatrix  $J(x) := [\alpha F_{\dot{x}} + F_x](\alpha x + \beta, x, \tau_{n+1})$  möglichst lange über mehrere Integrationsschritte hinweg unverändert gelassen und nur dann neu berechnet, wenn eine ausreichend schnelle Konvergenz der Iteration ausbleibt (modifiziertes Newton-Verfahren). Zudem soll sich eine geschickte Wahl des Faktors  $\gamma > 0$  konvergenzbeschleunigend auf die Iteration auswirken – Details zu den Werten von  $\gamma$  sind in [BCP96] nachzulesen. Wenn für die Berechnung der Jacobimatrix keine benutzerdefinierte Funktion bereitgestellt wird, dann approximiert DASPK3.1 die Matrix mit einseitigen Differenzenquotienten. Die  $i$ -te Spalte von  $J(x)$  wird mit  $x = (x_1, \dots, x_{n_x})^\top$  somit durch

$$\frac{F(\alpha x + \beta + \alpha \Delta x_i e_i, x + \Delta x_i e_i, \tau_{n+1}) - F(\alpha x + \beta, x, \tau_{n+1})}{\Delta x_i}$$

angenähert. Für das Inkrement wird  $\Delta x_i = \sqrt{\text{eps}} \cdot \max(|x_i|, |h_n(\alpha x_i + \beta_i)|, |x_i| \text{RTOL}_i + \text{ATOL}_i)$  gewählt.

Wie genau die berechnete numerische Lösung  $x$  sein soll, kann über die relativen und absoluten Fehlertoleranzen  $\text{RTOL}, \text{ATOL} \in \mathbb{R}^{n_x}$  vorgegeben werden. Es wird dann mit Hilfe der gewichteten 2-Norm

$$\|\varepsilon\| = \sqrt{\frac{1}{n_x} \sum_{i=1}^{n_x} \left( \frac{\varepsilon_i}{|x_i| \text{RTOL}_i + \text{ATOL}_i} \right)^2}$$

eines Fehlerschätzers  $\varepsilon = (\varepsilon_1, \dots, \varepsilon_{n_x})^\top$  versucht, das Abbruchkriterium für die Korrekturiteration und die Berechnung der Schrittweiten so zu steuern, dass die geforderten Toleranzvorgaben für den aktuell berechneten Wert  $x = x_{n+1}$  eingehalten werden. Auf der



anderen Seite sollen diese Vorgaben mit möglichst geringem Berechnungsaufwand realisiert werden. Dazu wählt der implementierte Algorithmus die Schrittweiten und die Ordnung  $1 \leq k \leq 5$  des Verfahrens automatisch unter Berücksichtigung des Lösungsverhaltens. Am Anfang der Integration ( $n = 1$ ) kann die Ordnung nur schrittweise von  $k = 1$  an erhöht werden, da stets  $k \leq n$  gelten muss, um genügend Interpolationspunkte für das BDF-Polynom zur Verfügung zu haben.

Das Polynom dient auch dazu, eine stetige Darstellung der numerischen Lösung zu erzeugen. Für beliebiges  $t \in [\tau_n, \tau_{n+1}]$  wird als numerische Approximation für  $x(t)$  der Wert des Polynoms verwendet, das zur Berechnung von  $x_{n+1}$  konstruiert wurde. Des Weiteren spielt das Interpolationspolynom eine wichtige Rolle, um die Nullstellen von Schaltfunktionen  $\rho(\dot{x}, x, t)$ ,  $\rho : \mathbb{R}^{2n_x+1} \rightarrow \mathbb{R}^{n_\rho}$  genau zu berechnen. Hierfür wird nach der erfolgreichen Berechnung von  $x_{n+1}$  geprüft, ob für  $i = 1, \dots, n_\rho$  die Werte  $\rho_i(\dot{x}_{n+1}, x_{n+1}, \tau_{n+1})$  und  $\rho_i(\dot{x}_n, x_n, \tau_n)$  unterschiedliche Vorzeichen haben. Wenn das für ein  $i$  der Fall ist, bestimmt DASPK3.1 ein Intervall  $[\tau^*, t^*] \subset [\tau_n, \tau_{n+1}]$ , für das die Bedingungen

$$\rho_{i^*}(\tau^*) \rho_{i^*}(t^*) \leq 0, \quad |t^* - \tau^*| < 100 \text{ eps} |\tau_{n+1}| \quad \text{und} \quad \rho_{i^*}(\tau^*) \neq 0$$

gelten. Der Index  $i^*$  wird dabei so gewählt, dass der Abstand  $|t^* - \tau_n|$  bzgl.  $i$  minimal ist. Für die Auswertungen von  $\rho$  wird zur Approximation von  $x$  und  $\dot{x}$  das BDF-Polynom und seine Ableitung herangezogen. Mit dem Test

$$\rho_i(\tau^*) \overset{?}{\rho_i(t^*)} \leq 0, \quad i = 1, \dots, n_\rho$$

werden nach der Berechnung von  $[\tau^*, t^*]$  alle Komponenten auf Nullstellen geprüft, um mehrere aktive Schaltfunktionen zu erkennen. Der Zeitpunkt  $t^*$  wird als Ereigniszeitpunkt der so erkannten aktiven Schaltfunktionen angesehen.

In DASPK3.1 ist zusätzlich zu den genannten Routinen ein Teil für die Integration der Sensitivitätsgleichungen enthalten. Im Nachfolgenden beschreiben wir diesen Teil genauer. Ein erster wichtiger Punkt betrifft die Auswertung der Sensitivitätsgleichungen, die für ein parameterabhängiges differential-algebraisches System

$$F(\dot{x}, x, p, t) = 0, \quad x(p, t^0) = A(p)$$

die folgende Gestalt annehmen:

$$F_{\dot{x}}(\dot{x}, x, p, t) \dot{x}_p + F_x(\dot{x}, x, p, t) x_p + F_p(\dot{x}, x, p, t) = 0, \quad x_p(p, t^0) = A_p(p).$$

Entweder werden die Ableitungen  $F_{\dot{x}}$ ,  $F_x$ ,  $F_p$  vom Benutzer bereitgestellt, z. B. auch über automatisches Differenzieren, oder finite Differenzen kommen zur Anwendung. Um Auswertungen von  $F$  einzusparen und dennoch eine hohe Genauigkeit zu erreichen, werden Richtungsableitungen zweiter Ordnung eingesetzt:

$$\frac{F(\dot{x} + \dot{x}_p \delta p_j, x + x_p \delta p_j, p + \delta p_j, t) - F(\dot{x} - \dot{x}_p \delta p_j, x - x_p \delta p_j, p - \delta p_j, t)}{2\Delta p_j} = 0. \quad (5.4)$$

Als abkürzende Bezeichnung benutzen wir  $\delta p_j := \Delta p_j e_j$  mit  $j = 1, \dots, n_p$ . Den Wert für  $\Delta p_j$  zu definieren, ist insbesondere für ungünstig skalierte Probleme schwierig. In

der Literatur gibt es unterschiedliche Ansätze,  $\Delta p_j$  möglichst gut zu wählen. In einer Vorgängerversion [MP96] von DASPK3.1 wird ein relatives Inkrement  $\mathcal{O}(\sqrt{\text{eps}})$  für den einseitigen Differenzenquotienten vorgeschlagen. Für den symmetrischen Differenzenquotienten wird in [LH00, SH03] ein relatives Inkrement  $\mathcal{O}(\sqrt{\text{RTOL}})$  empfohlen. In DASPK3.1 wird versucht, über die Größenordnungen von  $p_j$ ,  $x$  und  $x_{p_j}$  einen Kompromiss zu finden, der in [LP99] näher begründet wird:

$$\Delta p_j := \zeta \max(|p_j|, \|\vartheta^j\|_2^{-1}). \quad (5.5)$$

Die Vektoren  $\vartheta^j$  errechnen sich dabei aus

$$\vartheta^j = (\vartheta_1^j, \dots, \vartheta_{n_x}^j)^\top, \quad \vartheta_i^j := \frac{|(x_p)_{ij}| \overline{\text{RTOL}}_{ij} + \overline{\text{ATOL}}_{ij}}{|x_i| \text{RTOL}_i + \text{ATOL}_i}. \quad (5.6)$$

Die eingesetzten Fehlertoleranzen  $\overline{\text{RTOL}}, \overline{\text{ATOL}} \in \mathbb{R}^{n_x \times n_p}$  korrespondieren zu den Sensitivitätszuständen  $x_p$  und können individuell vorgegeben werden. Der Faktor  $\zeta > 0$  in (5.5) kann benutzerdefiniert vorgegeben werden, standardmäßig wird dafür der Wert  $10^{-3}$  vorgeschlagen. In Abschnitt 5.2.1 werden wir eine vom Standardwert abweichende Definition für  $\zeta$  erläutern.

Insgesamt erhalten wir  $n_x n_p$  Sensitivitätsgleichungen und  $n_x$  Nominalgleichungen, die numerisch gelöst werden müssen. Wie in Abschnitt 5.1.1 angedeutet, werden die Gleichungen gleichzeitig integriert, damit genaue Ergebnisse mit einer effizienten Implementierung berechnet werden können. In jedem Integrationsschritt führen die Gleichungen vektoriell zusammengefasst zu dem System

$$\Phi(\dot{X}, X, p, t) = 0, \quad X(p, t^0) = \left( A^\top(p), A_{p_1}^\top(p), \dots, A_{p_{n_p}}^\top(p) \right)^\top$$

mit den Bezeichnungen

$$X := \begin{pmatrix} x \\ x_{p_1} \\ \vdots \\ x_{p_{n_p}} \end{pmatrix}, \quad \Phi(\dot{X}, X, p, t) := \begin{pmatrix} F \\ F_{\dot{x}} \dot{x}_{p_1} + F_x x_{p_1} + F_{p_1} \\ \vdots \\ F_{\dot{x}} \dot{x}_{p_{n_p}} + F_x x_{p_{n_p}} + F_{p_{n_p}} \end{pmatrix} (\dot{x}, x, p, t).$$

Analog zu Gleichung (5.3) führt der BDF-Ansatz zu einem nichtlinearen Gleichungssystem

$$\Phi(\alpha X + B, X, p, \tau_{n+1}) = 0 \quad (5.7)$$

mit den Vektoren

$$B = \left( \beta^\top, B_1^\top, \dots, B_{n_p}^\top \right)^\top, \quad B_i = \sum_{j=1}^k \alpha_{n, k-j} x_{p_i, n+1-j} \quad (i = 1, \dots, n_p).$$

Zur Lösung dieses Systems kann in DASPK3.1 eine von drei Varianten gewählt werden. Die erste Technik betrachtet Gleichung (5.7) als Gesamtsystem, das mit dem modifizierten Newton-Verfahren

$$X^{(m+1)} = X^{(m)} - \gamma \Lambda^{-1}(X^{(m)}) \Phi(\alpha X^{(m)} + B, X^{(m)}, p, \tau_{n+1})$$

und der Jacobimatrix

$$\Lambda(X) = \begin{pmatrix} J & 0 & \cdots & \cdots & 0 \\ J_1 & J & 0 & & \vdots \\ J_2 & 0 & J & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ J_{n_p} & 0 & \cdots & 0 & J \end{pmatrix} (X)$$

mit

$$J(x) := \tilde{J}(\alpha x + \beta, x, p, \tau_{n+1}), \quad \tilde{J}(\dot{x}, x, p, t) := [\alpha F_{\dot{x}} + F_x](\dot{x}, x, p, t),$$

$$J_j(x, x_{p_j}) := \tilde{J}_j(\alpha x + \beta, \alpha x_{p_j} + B_j, x, x_{p_j}, p, \tau_{n+1})$$

sowie

$$\tilde{J}_j(\dot{x}, \dot{x}_{p_j}, x, x_{p_j}, p, t) := \frac{d}{d\dot{x}} \left( \tilde{J}(\dot{x}, x, p, t) \dot{x}_{p_j} \right) + \frac{d}{dx} \left( \tilde{J}(\dot{x}, x, p, t) x_{p_j} \right) + \tilde{J}_{p_j}(\dot{x}, x, p, t)$$

gelöst wird. Deswegen bezeichnen die Autoren diese Variante als *gleichzeitiges Korrektungsverfahren*. In der Implementierung reicht es für die lokale Konvergenz aus, die approximierte Jacobimatrix  $\text{blockdiag}(J, \dots, J)$  in einem modifizierten Newton-Verfahren zu verwenden [MP96]. Das erspart die evtl. aufwendige Berechnung der Matrizen  $J_j$  und ermöglicht die Beibehaltung von  $J$  über mehrere Integrationsschritte.

Der zweite Ansatz für die Lösung von Gleichung (5.7) sieht eine stufenweise Vorgehensweise vor. Zunächst wird wie für den Fall, dass keine Sensitivitäten berechnet werden, nur die Korrekturiteration für die Lösung  $x_{n+1}$  ausgeführt. Ausgehend von dieser gibt es zwei Möglichkeiten die Sensitivitäten zu ermitteln. Im *stufenweisen Korrektungsverfahren* iteriert ein modifiziertes Newton-Verfahren bzgl.  $x_{p_j}$  über den Gleichungen

$$[F_{\dot{x}}(\alpha x_{p_j} + B_j) + F_x x_{p_j} + F_{p_j}](\dot{x}_{n+1}, x_{n+1}, p, \tau_{n+1}) = 0. \quad (5.8)$$

Die dafür nötige Jacobimatrix  $J(x_{n+1})$  wird durch die Iterationsmatrix für das Nominalsystem ersetzt. Damit ist keine zusätzliche Auswertung und Zerlegung von  $J$  erforderlich. Will man den Aufwand aufbringen, die Matrix  $J$  in jedem Integrationsschritt zu berechnen und zu faktorisieren, dann werden die bzgl.  $x_{p_j}$  linearen Gleichungen (5.8) direkt gelöst, und man erhält das *stufenweise direkte Verfahren*.

## 5.2 FISEMO: Filippov-Lösungen und Sensitivitätsanalyse von Modelica-Modellen

In Abschnitt 3.1.3 und 3.2.1 zeigen Beispiele, welche unstetigen Modelle Filippov-Lösungen besitzen und dass die verallgemeinerten Lösungen die Dynamik der Systeme korrekt wiedergeben. Will man die entsprechenden Modellgleichungen mit einer Schaltlogik numerisch lösen, die nur die beiden Schalterpositionen  $\pm 1$  kennt, so führt dies zu sehr häufigem Hin- und Herschalten zwischen den Schalterzuständen [Eic92]. Der Effekt wird in der Literatur auch als *Chattering* [UGS99] bezeichnet. Praktisch bedeutet Chattering, dass die

numerische Integration sehr ineffizient wird und meistens mit einem benutzergewollten Abbruch endet. Folglich gehören Modelle mit Filippov-Lösungen nicht zu der Problemklasse, die mit einer zweiwertigen Schaltlogik effizient behandelt werden können.

Ein Ziel der im Rahmen der vorliegenden Arbeit neu entwickelten Umgebung ist die automatische Berechnung der Filippov-Lösung von Modelica-Modellen. Damit kann eine breitere Problemklasse von unstetigen Modellen ohne aufwendige Modelländerungen rechnerunterstützt gelöst werden.

Das zweite Ziel der Softwareumgebung ist die Ermittlung der Parametersensitivitäten – insbesondere für unstetige Modelle. Um ein möglichst universelles Werkzeug zu erhalten, wird die Integration der Sensitivitätsgleichungen als Berechnungsmethode gewählt. Die Integration der adjungierten Gleichungen liefert im Vergleich nur die Parameterableitungen ausgewählter Ausgabegrößen am Endzeitpunkt. Externe numerische Differentiation mit finiten Differenzen kann zu sehr ungenauen Resultaten führen und einen hohen Rechenbedarf erfordern. Wie in Abschnitt 5.1.1 dargelegt, ist die interne numerische Differentiation eng mit der numerischen Lösung der Sensitivitätsgleichungen verwandt, lediglich Implementierungsdetails können unterschiedlich sein.

Mit DASPK3.1 verwenden wir einen frei verfügbaren Code<sup>1</sup>, der die Sensitivitätsgleichungen mittels BDF-Verfahren mit Schrittweiten- und Ordnungssteuerung numerisch integriert (siehe vorhergehenden Abschnitt). Grundlage hierfür ist DASSL, das in Dymola eingebettet einen der wichtigsten Integrationsalgorithmen zur numerischen Behandlung von Modelica-Modellen darstellt. Das Mehrschrittverfahren berechnet besonders für steife Probleme effizient und robust die Lösungstrajektorien.

Enthalten die Trajektorien viele Ereigniszeitpunkte, dann können Einschrittverfahren effizienter sein, weil diese Verfahren nach dem Ereignispunkt schneller starten und dann dort mit einer höheren Ordnung rechnen. Es existieren auch Ansätze, Mehrschrittverfahren mit einigen Schritten eines Einschrittverfahrens zu starten [Gea80, SB95]. Diese wurden nach Wissen des Autors bisher nicht zur Integration von Modelica-Modellen eingesetzt. Die Wahl der Anfangsschrittweite im Anschluss an einen Ereignispunkt kann sich ebenfalls wesentlich auf die Rechenzeit der Integration auswirken, wenn zahlreiche Ereignisse eintreten. Erlauben es die Ereignisse, die (großen) aktuellen Schrittweiten, die kurz vor den Ereigniszeitpunkten möglich waren, beizubehalten, so beschleunigen entsprechende Anfangsschrittweiten die Startphase nach den Ereigniszeitpunkten [KO06].

In den nächsten beiden Abschnitten stellen wir die entwickelte Software-Umgebung FISEMO vor, die es einheitlich ermöglicht, die Filippov-Lösungen von unstetigen multidisziplinären Modellen zu ermitteln. Des Weiteren werden innerhalb der Umgebung gleichzeitig mit der Lösung die Sensitivitätstrajektorien der unstetigen Modelle berechnet.

### 5.2.1 Die Softwareumgebung FISEMO

Die Softwareumgebung FISEMO zur Berechnung von Filippov-Lösungen und zur numerischen Sensitivitätsanalyse von Modelica-Modellen ist modular aufgebaut. Sie besteht im

---

<sup>1</sup>Der Autor ist L. R. Petzold dankbar für die Überlassung des Codes DASPK3.1.

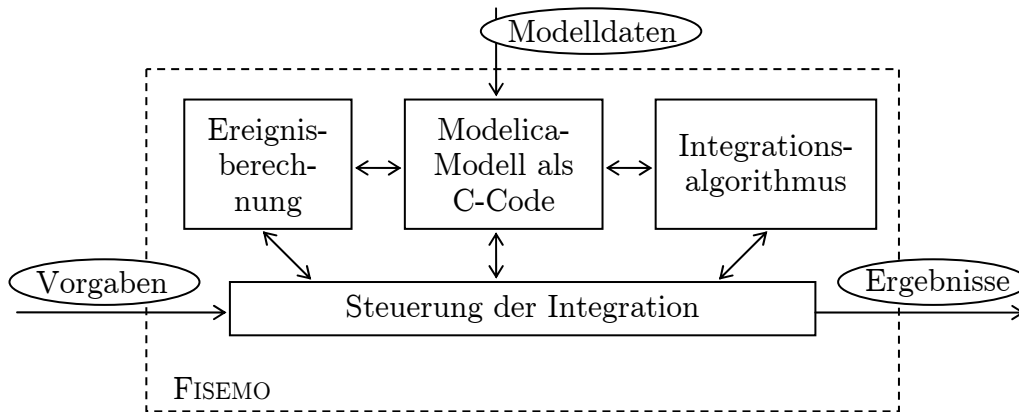


Abbildung 5.1: Programmarchitektur von FISEMO zur Berechnung von Filippov-Lösungen und zur numerischen Sensitivitätsanalyse von Modelica-Modellen.

Wesentlichen aus den in Abbildung 5.1 gezeigten vier Einheiten Modell, Integrationsalgorithmus, Ereignisberechnung und Steuerung der Integration<sup>2</sup>. Wir beschreiben in diesem Abschnitt das Zusammenwirken des erzeugten Modellcodes mit dem Integrationspaket DASPK3.1. Die Steuerung der Integration und die Berechnungen am Ereignis sind dann Gegenstand des nächsten Abschnitts.

Vom Anwender werden die Anforderungen vorgegeben, bzgl. welcher Modellparameter  $p$  eine Sensitivitätsanalyse durchgeführt werden soll und welche Ausgabegrößen  $y$  nach Parametern differenziert werden sollen. Alternativ können auch die Sensitivitäten aller Zustände  $x$  ausgegeben werden. Die Ergebnisse werden auf einem konstanten Zeitgitter mit vorzugebender Gitterweite ausgegeben, das von den Ereigniszeitpunkten überlagert wird. Zudem sind das Integrationsintervall  $[t^0, t^\infty]$  und sämtliche Toleranzvorgaben  $\text{RTOL}$ ,  $\text{ATOL}$ ,  $\overline{\text{RTOL}}$ ,  $\overline{\text{ATOL}}$  der Integration zu definieren.

Als Modell verstehen wir den von Dymola [Dym] automatisch erzeugten C-Code eines Modelica-Modells. Dieser Code entspricht dem in Abschnitt 2.2.2 erläuterten transformierten Modell in numerischer Zustandsform. Die Schnittstelle zwischen dem Modell und seinen aufrufenden Funktionen besteht neben den Modellvariablen aus einem Modusparameter `idemand`, der verschiedene Auswertungsmoden ermöglicht, die in Tabelle 5.1 aufgelistet sind.

Vor der ersten Auswertung des Modells muss dieses im Modus `idemand = 0` aufgerufen werden, um die externen Modelldaten zu laden und alle internen Variablen, insbesondere auch die Schalterpositionen von unstetigen Modellen, zu initialisieren. Die Moden 1–4 unterscheiden sich dahingehend, dass bei festen Schalterpositionen je nach Modus nur bestimmte Teile aller Modellgleichungen ausgewertet werden. Dieser Ansatz steigert die Effizienz der Modellauswertungen. An Ereignispunkten muss das Modell im Modus `idemand = 5` ausgewertet werden, damit alle Schalter und Variablen richtig gesetzt

<sup>2</sup>Der Autor bedankt sich bei H. Olsson (Dynasim AB, Lund) für eine Musterschnittstelle zwischen dem C-Code eines Modelica-Modells und der Integrationsroutine DASKR.

Tabelle 5.1: Auswertungsmoden eines von Dymola transformierten Modelica-Modells.

idemand	Modus
0	Anfangsinitialisierung zur Zeit $t^0$ .
1	Berechnung der relevanten Ausgabegrößen $y = g(x, p, t)$ .
2	Auswertung der rechten Seite $f(x, p, t)$ .
3	Berechnung aller algebraischen Variablen im Modell.
4	Auswertung der Schaltfunktionen $Q(x, p, t)$ .
5	Neuinitialisierung an einem Ereignispunkt.

werden. Diese Berechnung führt modellintern oft zu mehreren Modellauswertungen, da die diskreten Variablen und Schalterpositionen iterativ bestimmt werden, vgl. Iteration 2.8.

Die Schalterpositionen nehmen jeweils einen der zwei möglichen Werte  $\pm 1$  an, die in unmittelbarem Zusammenhang mit den Wahrheitswerten der zugehörigen Relationen im Modell stehen. Grundlage der unstetigen Modelle ist somit zunächst eine zweiwertige Schaltlogik, die um den dritten Wert 0 außerhalb des eigentlichen Modells ergänzt wird. Dazu liegt zwischen dem Modell und der Integratorschnittstelle zur Auswertung von  $F$  und  $\rho$  die Prozedur `evalModel`. Diese stellt für den Fall, dass eine Filippov-Lösung berechnet werden soll, die entsprechende rechte Seite durch zweimaliges Auswerten des Modells mit den beiden Schalterstellungen der Schaltfunktion  $q$  bereit. Ebenso werden die Ersatzschaltfunktionen  $D^+q$ ,  $D^-q$  berechnet.

Die allgemeine Integratorschnittstelle für die Berechnung von  $F$  und  $\rho$  ist in DASPK3.1 mit den beiden Funktionen `res` und `rt` vorgegeben. Da die transformierten Modelica-Modelle in numerischer Zustandsform verfügbar sind, wird aus Sicht des Integrationsverfahrens ein Index-0-System mit  $F(\dot{x}, x, p, t) := \dot{x} - f(x, p, t)$  und  $\rho(\dot{x}, x, p, t) := Q(x, p, t)$  numerisch gelöst. Diese Umrechnung erfolgt in den Funktionen `res` und `rt`, nachdem die rechte Seite bzw. die Schaltfunktionen von `evalModel` ausgewertet wurden. Wenn keine Filippov-Lösung vorliegt, dann werden die Funktionsargumente direkt an die Modellauswertung weitergereicht – `evalModel` dient somit als Hülle der Auswertung.

Der Festlegung des Inkrementfaktors  $\zeta$  in Gleichung (5.5) kommt besondere Bedeutung zu, da von  $\zeta$  auch die Genauigkeit der numerischen Sensitivitäten abhängen kann. Der Faktor geht in die finite Differenzenapproximation (5.4) bei der Auswertung der Sensitivitätsgleichungen ein. Die numerische Jacobimatrix hingegen trägt bei einer Nominalintegration ohne Sensitivitätsgleichungen vor allem zu einer effizienten Lösung der impliziten Gleichungssysteme in jedem Integrationsschritt bei. Wir definieren

$$\zeta := \sqrt[3]{\text{eps}_f}, \quad (5.9)$$

das nach einer analogen Fehlerbetrachtung wie in Abschnitt 5.1.1 für Differenzenquotienten zweiter Ordnung optimal ist. Die Genauigkeit  $\text{eps}_f$  gibt dabei die Genauigkeit der rechten Seite  $f$  wieder. In den Sensitivitäten ergibt sich damit ein Fehler der Größenordnung

$$\mathcal{O}(\text{TOL}) + \mathcal{O}(\Delta p_j^2) + \mathcal{O}\left(\frac{\text{eps}_f}{\Delta p_j}\right) = \mathcal{O}(\text{TOL}) + \mathcal{O}\left(\text{eps}_f^{2/3}\right), \quad (5.10)$$

wenn  $\text{TOL} = \text{RTOL}_i = \text{ATOL}_i = \overline{\text{RTOL}}_{ij} = \overline{\text{ATOL}}_{ij}$  gilt. Folglich ist die Genauigkeit der numerischen Sensitivitäten in der Größenordnung der Genauigkeit der Zustände, falls  $\text{TOL} \geq \text{eps}_f^{2/3}$  erfüllt ist. Weiter wählen wir für die Berechnung der Sensitivitäten das stufenweise Korrektungsverfahren, da es das größte Effizienzpotential bietet.

### 5.2.2 Numerische Integration von unstetigen Modelica-Modellen

Der Ablauf der numerischen Integration eines Modelica-Modells innerhalb von FISEMO ist in Anhang A, Abbildung A.1 dargestellt. Zu Beginn der Integration bei  $t^0$  wird das Modell initialisiert, und es werden die Sensitivitäten

$$x_{p_j}(p^*, t^0) = A_{p_j}(p^*) = \frac{A(p^* + \Delta p_j e_j) - A(p^* - \Delta p_j e_j)}{2\Delta p_j} + \mathcal{O}(\Delta p_j^2) \quad (5.11)$$

der Anfangswerte berechnet. Am Anfang der Integrationschleife werden zunächst die Sensitivitäten  $y_p$  der Ausgabegrößen gemäß der Diskretisierung

$$y_{p_j}(p^*, t) = \frac{g(\eta_j^+(p^*, t)) - g(\eta_j^-(p^*, t))}{2\Delta p_j} + \mathcal{O}(\Delta p_j^2) \quad (5.12)$$

mit den Argumenten

$$\eta_j^\pm(p, t) := (x(p, t) \pm x_{p_j}(p, t) \Delta p_j e_j, p \pm \Delta p_j e_j, t) \quad (5.13)$$

berechnet und anschließend zusammen mit den anderen Größen an eine Ergebnisroutine gemeldet, die die Speicherung und Ausgabe der Resultate übernimmt. Diese fasst die angemeldeten Ergebnisse mit früheren zusammen, wenn sie nicht genügend zeitlichen Abstand zueinander haben. Dies wird erforderlich, falls Ereigniszeitpunkte sehr dicht beieinander oder zufällig sehr nah an Ausgabezeitpunkten liegen.

Im folgenden Schritt bestimmt der Algorithmus den nächsten Zeitpunkt, an dem die Integrationsprozedur Ergebnisse zurückliefern soll. Zur Auswahl stehen der nächste reguläre Ausgabezeitpunkt auf dem konstanten Zeitgitter, das nächste Zeitereignis oder der Endzeitpunkt  $t^\infty$ . Als *Zeitereignis* (englisch: time event) bezeichnen wir dabei ein Ereignis, dessen Eintrittszeitpunkt im Vorhinein bekannt ist. Es korrespondiert zu einer Schaltfunktion der Form  $q^{\text{ZE}}(p, t) = T(p) - t$ , für die keine Nullstellensuche im Integrationsalgorithmus nötig ist, da  $q^{\text{ZE}} = 0$  einfach auflösbar und  $t^* = T(p^*)$  nicht zustandsabhängig ist. Jedes Ereignis, das kein Zeitereignis ist, heißt auch *Zustandsereignis* (englisch: state event).

Jetzt sind alle Vorbereitungen getroffen, um dem Integrationsalgorithmus die numerische Berechnung der Lösung  $x$  und ihrer Sensitivitäten  $x_p$  zu überlassen. DASPK3.1 liefert die Werte hierfür entweder zum definierten Ausgabezeitpunkt oder zum ersten Zustandsereignis, das vorher stattfindet, zurück. Die Indizes der aktiven Schaltfunktionen stehen nach der Rückgabe ebenfalls zur Verfügung. Wenn keine Schaltfunktion aktiv ist, also bis zum Ausgabezeitpunkt kein Ereignis eingetreten ist, dann kann direkt zurück zum Anfang der Integrationschleife gesprungen werden. Andernfalls werden die Sensitivitäten der Ausgabegrößen zum aktuellen Rückgabezeitpunkt berechnet und alle Ergebnisse zur



Ausgabe angemeldet. Es folgen die Berechnungen am Ereignispunkt, die auf den nächsten Seiten genauer erläutert werden. Sind die Ereignisberechnungen abgeschlossen, so wird der Integrationsablauf, beginnend mit der Berechnung der Sensitivitäten für die Ausgabegrößen, von Neuem durchlaufen. Die Durchläufe wiederholen sich solange, bis das Ende des Integrationsintervalls überschritten würde.

### Bemerkung 5.1 (Voraussetzungen an das unstetige Modelica-Modell)

Die theoretischen Untersuchungen in den Kapiteln 3 und 4 zeigen, dass die Modellstruktur eines hybriden Modells wesentlichen Einfluss darauf hat, ob (Filippov-)Lösungen und gegebenenfalls Sensitivitäten des entsprechenden Systems existieren. Daher verwundert es nicht, wenn wir für die numerische Berechnung Anforderungen an die Unstetigkeiten eines Modelica-Modells stellen:

- a) Wir setzen voraus, dass keine unstetigen Zustandsänderungen an Ereignispunkten vorkommen:  $A^i = \text{id}_x$  für  $i \geq 1$ .
- b) Das hybride System lasse sich an allen Ereignispunkten wie in Korollar 3.27 schreiben. Reguläre Schaltfunktionen seien zugelassen.
- c) Alle schalterabhängigen Schaltfunktionen  $Q^i$  dürfen höchstens vom eigenen Schalter  $S_i$  und nicht von zusätzlichen Schaltern abhängen:  $Q^{i,S} = Q^{i,S_i}$ .

Die Voraussetzung a) erscheint auf den ersten Blick eine einschneidende Einschränkung der Problemklasse zu sein. Aus praktischer Sicht ist dies nicht der Fall, da unstetige Zustandsänderungen in anwendungsrelevanten Modelica-Modellen eher selten vorkommen. Wir nehmen die Einschränkung vor, weil mit dem von Dymola erzeugten C-Code des Modells keine wiederholten Auswertungen von  $A^i$  mit gestörten Argumenten möglich sind. Diese wären aber für die Berechnung von finiten Differenzen nötig.

In Satz 4.28 wird für Systeme mit einer speziellen Struktur und gleichzeitig aktiven Schaltfunktionen die Existenz der Sensitivitäten nachgewiesen. Da Informationen, wie sie für diese Struktur und die Behandlung von schalterabhängigen Schaltfunktionen im Allgemeinen benötigt werden, nicht im C-Code des Modells zur Verfügung stehen, passen wir mit den Voraussetzungen b) und c) die Problemklasse entsprechend an. Strukturdaten können nur während der Umwandlung des ursprünglichen Modelica-Modells in die numerische Zustandsform gewonnen werden, was aber im Rahmen der Arbeit nicht weiterverfolgt wurde.

Mit dem System aus Korollar 3.27 sind unabhängige Berechnungen für jede von mehreren gleichzeitig aktiven Schaltfunktionen möglich. Insbesondere die Reihenfolge der aktiven Schaltfunktionen spielt dabei keine Rolle. Satz 4.28 sichert unter diesen Voraussetzungen zusammen mit Bemerkung 4.29 die Existenz der Sensitivitäten  $\bar{x}_p$  und deren unabhängige Berechnung am Ereignispunkt:

$$\bar{x}_p(p^*, t^*) = x_p(p^*, t^*) + \sum_{k=1}^{n_q} [f^{k-1} - f^k](x^*, p^*, t^*) t_p^k(p^*). \quad (5.14)$$

Dabei sei  $f^k := f^{k-}$  mit  $f^{k-}$  aus Gleichung (4.11) auf Seite 62. □



Abbildung A.2 zeigt die Berechnungslogik an Ereignispunkten, die sich in zwei wesentliche Teile gliedert. Zunächst werden die Indizes der aktiven Schaltfunktionen ermittelt, in deren Schaltflächen die Lösung nach dem Ereignispunkt verbleibt. Die Untersuchung der aktiven Schaltfunktionen wird gemäß Kapitel 3.1 umgesetzt. Da wegen der Voraussetzungen aus Bemerkung 5.1 unabhängige Betrachtungen für jede aktive Schaltfunktion möglich sind, berechnen wir für alle Modellschaltfunktionen  $q^i$  mit  $q^i(x^*, p^*, t^*) = 0$  die Richtungsableitung

$$D^+q^i = q_x^i f^{s(i+)} + q_t^i$$

über die Diskretisierung

$$D^+q^i(x^*, p^*, t^*) = \frac{q^i(\mu_i^+(x^*, p^*, t^*)) - q^i(\mu_i^-(x^*, p^*, t^*))}{2\Delta t} + \mathcal{O}(\Delta t^2) \quad (5.15)$$

mit

$$\mu_i^\pm(x, p, t) := (x \pm f^{s(i+)}(x, p, t) \Delta t, p, t \pm \Delta t).$$

Analog werde  $D^-q^i$  bestimmt. Die rechte Seite  $f^{s(i+)}$  entsteht aus  $f^s$  durch Umschalten des Schalters  $s_i$  auf  $+1$ , vgl. auch Korollar 3.27. Schaltfunktionen, bzgl. derer Filippov-Lösungen lokal für  $t \leq t^*$  konstruiert wurden, sind in diesen Berechnungen enthalten. Aktive Ersatzschaltfunktionen einer bisherigen Filippov-Lösung sind ebenso wie  $q^{\text{ZE}}$  von den Berechnungen ausgenommen. Aktive Ersatzschaltfunktionen zeigen an, dass eine Filippov-Lösung das Ende ihres Definitionsbereichs erreicht hat und die Lösung die Schaltfläche wieder verlässt. Da  $q^{\text{ZE}}$  nicht zustandsabhängig ist, gilt  $D^+q^{\text{ZE}} = D^-q^{\text{ZE}} = q_t^{\text{ZE}} = -1$ . Folglich kann die Schaltfunktion eines Zeitereignisses keine Filippov-Lösung erzeugen.

Für die Detektion einer Filippov-Lösung nehmen wir an, dass der konstruierte Fall 3 aus Tabelle 3.1 nicht auftritt. Der Fall könnte ohnehin nur bei  $t^0$  entstehen, wenn die Anfangswerte entsprechend gesetzt würden. Da die Information, welche Schalterstellung welches Vorzeichen von  $q^i$  repräsentiert, meistens fehlt, wird der Index der Schaltfunktionen daher bereits für eine Filippov-Lösung vorgemerkt, wenn die Vorzeichen der Richtungsableitungen unterschiedlich sind.

Im zweiten Teil des Ablaufplans aus Abbildung A.2 werden die Sensitivitätsberechnungen für den Ereignispunkt durchgeführt. Ziel ist es, die Sensitivitätszustände neu zu berechnen. Dazu nutzen wir in der Implementierung Formel (5.14), da man die Vektoren  $f^k$  durch sukzessives Umschalten der Schalter auf die Stellungen nach dem Ereignispunkt mit anschließendem Auswerten des Modells (`idemand` = 2) erhält. Zudem berechnen wir

$$t_p^k(p^*) = - \left[ \frac{q_x^k x_p(p^*, t^*) + q_p^k}{q_x^k f^s + q_t^k} \right] (x^*, p^*, t^*) \quad (5.16)$$

für  $k = 1, \dots, n_q$  mit der Diskretisierung

$$t_{p_j}^k(p^*) = - \frac{\Delta t}{\Delta p_j} \frac{q^k(\eta_j^+(p^*, t^*)) - q^k(\eta_j^-(p^*, t^*))}{q^k(\mu^+(x^*, p^*, t^*)) - q^k(\mu^-(x^*, p^*, t^*))} + \mathcal{O}(\Delta p_j^2) + \mathcal{O}(\Delta t^2), \quad (5.17)$$

die den Zähler und Nenner von Gleichung (5.16) durch Richtungsableitungen zweiter Ordnung approximiert. Als abkürzende Bezeichnungen seien dabei (5.13) und

$$\mu^\pm(x, p, t) := (x \pm f^s(x, p, t) \Delta t, p, t \pm \Delta t)$$

verwendet. Für aktive Ersatzschaltfunktionen muss  $t_p$  nicht berechnet werden, da die zugehörige Differenz  $f^{k-1} - f^k$  nach Beispiel 4.22 verschwindet.

Die Ableitung  $t_p^{\text{ZE}}$  des Zeitpunkts eines Zeitereignisses muss bereits beim letzten Ereignis-punkt berechnet worden sein, da  $q^{\text{ZE}}$  im Rahmen einer Modellauswertung mit `idemand` = 5 den Zeitpunkt  $T(p^*) > t^*$  des *nächsten* Zeitereignisses zurückliefert. Zur Approximation von  $t_p^{\text{ZE}}$  dient der Differenzenquotient zweiter Ordnung:

$$t_p^{\text{ZE}}(p^*) = T_p(p^*) = \frac{T(p^* + \Delta p_j e_j) - T(p^* - \Delta p_j e_j)}{2\Delta p_j} + \mathcal{O}(\Delta p_j^2). \quad (5.18)$$

Nachdem anschließend alle Schalter im Modell auf ihre richtige Position für die Zeit nach dem Ereignis gebracht werden, sind alle Berechnungen am Ereignispunkt abgeschlossen und die Integratorsteuerung kann die Zeitintegration mit der neuen rechten Seite fortführen.

### Bemerkung 5.2 (Inkrement in finiten Differenzen)

Die in diesem Abschnitt aufgestellten Differenzenquotienten zweiter Ordnung enthalten die Inkremente  $\Delta t$  und  $\Delta p_j$ , die wir in der Implementierung wie folgt festlegen:

$$\Delta t = \sqrt[3]{\text{eps}_M} |t^\infty - t^0|, \quad \Delta p_j = \sqrt[3]{\text{eps}_M} \max(|p_j|, \sigma_j).$$

Der Faktor  $\sqrt[3]{\text{eps}_M}$  resultiert aus analogen Überlegungen zum zentralen Differenzenquotienten wie im vorherigen Abschnitt. Die Größe  $\text{eps}_M$  definiert die Genauigkeit einer Modellauswertung, die wir in unseren Berechnungen mit  $\text{eps}_M = 10^{-16}$  festsetzen. In der Phase einer Filippov-Lösung erhält man damit eine geschätzte Genauigkeit der konstruierten rechten Seite von  $\text{eps}_f = \text{eps}_M^{2/3}$ . Aus diesen Überlegungen ergibt sich dann in Gleichung (5.5) und (5.9) für das relative Inkrement

$$\zeta = \sqrt[3]{\text{eps}_f} = \text{eps}_M^{2/9} \doteq 2.8 \cdot 10^{-4},$$

das wir als feste Größe auch außerhalb von Filippov-Lösungen einsetzen. Um zu kleine Werte für  $\Delta p_j$  zu verhindern, setzen wir für die Formeln (5.12), (5.17)  $\sigma_j := \|\vartheta^j\|_2^{-1}$  mit  $\vartheta^j$  aus Gleichung (5.6). In den restlichen Quotienten (5.11) und (5.18) sei  $\sigma_j := \sqrt[3]{\text{eps}_M}$  als untere Schranke gewählt.  $\square$

### Bemerkung 5.3 (Algebraische Variablen und Filippov-Lösung)

Wir weisen noch auf eine Randerscheinung hin, die während der Berechnung einer Filippov-Lösung (bzgl. einer Schaltfunktion  $q$ ) auftritt. In dieser Phase ist es unter Umständen nicht möglich, alle algebraischen Variablen im Modell eindeutig zu berechnen. Hängt die Funktion  $g = g^s$  zur Berechnung der algebraischen Variablen von der Schalterstellung  $s$  der Schaltfunktion  $q$  ab, so müsste  $g^s$  mit  $s = 0$  berechnet werden. Da vonseiten der Modellierung aber nur die zweiwertige Schaltlogik mit  $s = \pm 1$  berücksichtigt wird, ist dies nicht möglich. Man erhält für  $y(p, t)$  entweder  $g^{+1}(x(p, t), p, t)$  oder  $g^{-1}(x(p, t), p, t)$ . Dieses prinzipielle Problem könnte nur durch zusätzliche Sprachelemente im Modell gelöst werden, um  $g^s$  mit  $s = 0$  festzulegen.  $\square$

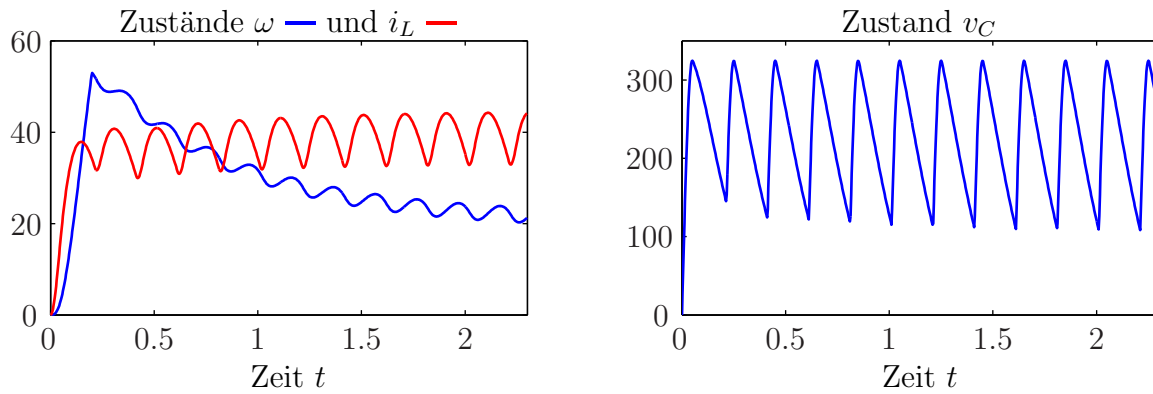


Abbildung 5.2: Zeitverläufe der drei Zustände für das Modell des belasteten Elektromotors, siehe Beispiel 5.4.

## 5.3 Numerische Ergebnisse

### Beispiel 5.4 (Belasteter Elektromotor – numerische Tests)















Wir unterziehen das Modelica-Modell aus Abbildung 2.2 auf Seite 11 einer Sensitivitätsanalyse und vergleichen die Ergebnisse der hier neu entwickelten Umgebung FISEMO mit den Resultaten einer externen numerischen Differentiation mit finiten Differenzen. Die Parameterwerte des Modells sind Tabelle 3.3 auf Seite 26 zu entnehmen. Für die Testrechnungen definieren wir das Lastdrehmoment wie in Beispiel 3.28. Die Einschaltzeit  $\tau = 0.2$  s ist so gewählt, dass sie nicht mit einem der Ereigniszeitpunkte, die von der Diode verursacht werden, zusammenfällt. Im Parametervektor  $p = (R_i, R, C, L, k, J, \bar{M}, \tau)^T$  fassen wir die Größen zusammen, bezüglich derer Sensitivitäten berechnet werden. Zunächst sind in Abbildung 5.2 die in FISEMO berechneten Zeitverläufe der drei Zustände  $x = (v_C, i_L, \omega)^T$  dargestellt. Der Spulenstrom  $i_L$  und die Kondensatorspannung  $v_C$  schwingen im Wesentlichen mit der Frequenz der Spannungsquelle jeweils um einen konstanten Wert. Die Drehgeschwindigkeit  $\omega$  des Motors nimmt bis zum Zuschalten der Last bei  $t = 0.2$  s monoton zu. Im Anschluss daran sinkt sie etwas ab und behält einen oberwelligen Anteil mit der Frequenz der Spannungsquelle.

In einer Serie von Testrechnungen untersuchen wir beispielhaft die Sensitivitäten der Winkelgeschwindigkeit  $\omega$ . Dazu integrieren wir mit DASPK3.1 das Nominalmodell zusammen mit den Sensitivitätsgleichungen für verschiedene Toleranzvorgaben  $\text{TOL}_{\text{SI}} := \text{RTOL}_i = \text{ATOL}_i = \overline{\text{RTOL}}_{ij} = \overline{\text{ATOL}}_{ij}$  ( $i = 1, \dots, n_x, j = 1, \dots, n_p$ ) im Bereich von  $10^{-3}$  bis  $10^{-9}$ . Zum Vergleich berechnen wir externe numerische Differenzenquotienten

$$\frac{\omega(p^* + \Delta p_j e_j, t) - \omega(p^*, t)}{\Delta p_j}$$

erster Ordnung mit  $\Delta p_j = p_j^* \sqrt{\text{TOL}_{\text{FD}}}$  wie in Abschnitt 5.1.1. Die verwendete Toleranzvorgabe  $\text{TOL}_{\text{FD}}$  bezieht sich auf die Integration des Nominalsystems mit den vorgegebenen und den gestörten Parameterwerten. Diese Integrationen werden ebenfalls mit FISEMO durchgeführt. In Anhang B sind in den Abbildungen B.3 und B.4 die Fehler der berechneten Sensitivitäten über der Zeit aufgetragen. Die linken Spalten der Abbildungen zeigen

Tabelle 5.2: Legende zu den Ergebnisabbildungen. SI: Sensitivitätsintegration. FD: Externe numerische Differentiation mit finiten Differenzen.

TOL	$10^{-3}$	$10^{-4}$	$10^{-5}$	$10^{-6}$	$10^{-7}$	$10^{-8}$	$10^{-9}$
SI							
FD							

jeweils die Ergebnisse der Sensitivitätsintegration (SI), wohingegen in den rechten Diagrammen die Fehler der externen Differenzenquotienten (FD) dargestellt sind. Die beiden Diagramme in jeder Zeile beziehen sich jeweils auf dieselbe Komponente des Parametervektors  $p$ . Die in Tabelle 5.2 aufgelisteten Farben symbolisieren die Toleranzvorgaben für die numerische Integration.

Die numerischen Fehler der Sensitivitätsverläufe sind als skalierte Fehler bezüglich einer Referenzlösung  $\omega_{p_j}^{\text{Ref}}$  dargestellt:

$$\frac{|\omega_{p_j}^{\text{TOL}}(t) - \omega_{p_j}^{\text{Ref}}(t)|}{1 + \bar{\omega}_{p_j}}, \quad \bar{\omega}_{p_j} := \frac{1}{N+1} \sum_{i=0}^N |\omega_{p_j}^{\text{Ref}}(t^0 + i \delta t)|, \quad N := \left\lfloor \frac{t^\infty - t^0}{\delta t} \right\rfloor. \quad (5.19)$$

Der zeitliche Mittelwert  $\bar{\omega}_{p_j}$  der absoluten Referenzlösung ergibt sich hierbei als arithmetisches Mittel der  $N+1$  Werte auf einem äquidistanten Zeitgitter mit der Gitterweite  $\delta t := 3 \cdot 10^{-3}$ . Die Skalierung bewirkt, dass für großes  $\bar{\omega}_{p_j}$  der relative Fehler und für kleine Werte von  $\bar{\omega}_{p_j}$  der absolute Fehler zum Tragen kommen. Als Referenzlösung dient die numerische Lösung mit  $\text{TOL}_{\text{SI}} = 10^{-9}$ .

Deutlich zu erkennen ist in den Abbildungen B.3 und B.4, dass die numerische Lösung sowohl von SI als auch von FD fast durchgängig genauer wird, wenn die Toleranzvorgaben kleiner gewählt werden. Vergleicht man die Ergebnisse in den linken Diagrammen der Abbildungen mit denen der rechten, so ist offensichtlich, dass (bei üblichen Fehlertoleranzen) eine wesentlich höhere Genauigkeit der Sensitivitäten mit SI erreicht werden kann als mit FD. Um dieselbe Genauigkeit zu erzielen wie mit  $\text{TOL}_{\text{FD}} = 10^{-9}$ , reicht sogar  $\text{TOL}_{\text{SI}} = 10^{-3}$  bzw.  $10^{-4}$  aus.

Eine detaillierte Bewertung, wie sich die Genauigkeitsvorgaben auf die tatsächliche Genauigkeit bei diesem Beispiel auswirken, zeigen die Abbildungen B.5 und B.6, die die Fehlerdiagramme mit Bezug auf den zu erwartenden Fehler  $\text{TOL}_{\text{SI}}$  bzw.  $\sqrt{\text{TOL}_{\text{FD}}}$  enthalten. Für SI bleiben die Sensitivitätsfehler der Parameter 3, 5, 7 und 8 sehr gut unter den vorgegebenen Schranken – z. T. sogar um ein bis zwei Größenordnungen. Die Fehler bzgl. Parameter 1, 2, 4, 6 bleiben zum Großteil unter  $10 \text{ TOL}_{\text{SI}}$ . Schwierigkeiten bereiten offenbar die kleinen Toleranzansprüche, was besonders bei Parameter 1 und 4 gilt.

Die Güte der externen Differenzenquotienten ist uneinheitlicher, da nur die Fehler der FD-Sensitivitäten von Parameter 3 und 7 unterhalb von  $\sqrt{\text{TOL}_{\text{FD}}}$  bleiben. Parameter 1 bereitet wie bei SI Schwierigkeiten, die allerdings darin resultieren, dass die ohnehin schon groben Fehlerschranken teilweise um mehrere Größenordnungen überschritten werden. Am ehesten erfüllen die Sensitivitäten mit den Toleranzen  $\text{TOL}_{\text{FD}} = 10^{-8}$ ,  $10^{-9}$  bei Parameter 1 die Vorgaben, wohingegen die Fehler für  $\text{TOL}_{\text{FD}} = 10^{-3}$ ,  $10^{-4}$  und  $10^{-5}$  zu

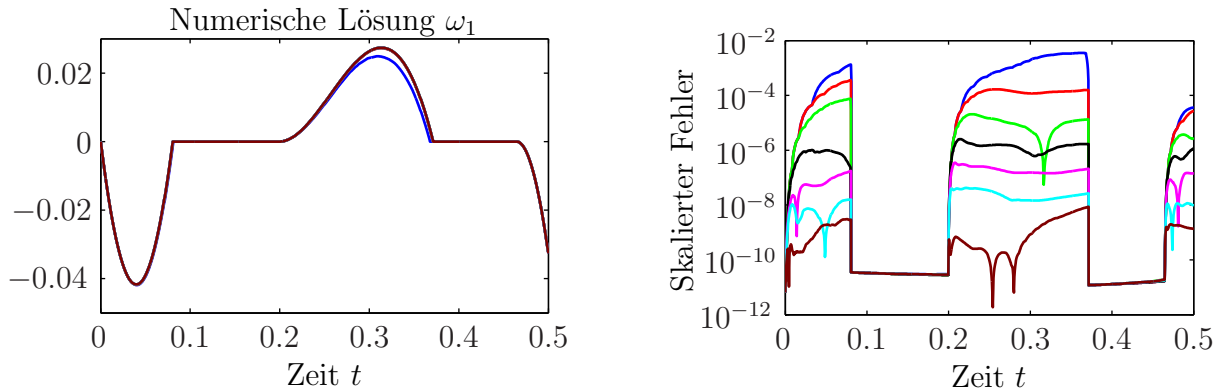


Abbildung 5.3: Numerische Lösungen (links) und zugehörige Fehler (rechts) der Winkelgeschwindigkeit  $\omega_1$  aus Beispiel 5.5 für verschiedene Fehlertoleranzen gemäß Tabelle 5.2.

deutlichen Abweichungen in den Zeittrajektorien der Sensitivitäten führen, siehe Abbildung B.7. Die Lösungskurven der restlichen Sensitivitäten sind optisch nur für die größeren Toleranzen von der Referenzlösung zu unterscheiden.

Die Lösung  $\omega_{ps} = \omega_\tau$  zeigt für  $t = 0.2$ s einen unstetigen Verlauf, der von den externen Differenzenquotienten nur näherungsweise (mit stetigen Kurven) aufgelöst wird. Daher ist im zugehörigen Diagramm in Abbildung B.6 an der Stelle  $t = 0.2$ s ein sprunghafter Ausschlag im Fehlerdiagramm sichtbar. Der unstetige Sensitivitätsverlauf wird durch die Berechnungen am Ereignispunkt, wie sie in Abschnitt 5.2.2 beschrieben sind, korrekt erfasst. Die Höhe des Sprungs wird unabhängig von  $TOL_{SI}$  über die unstetige rechte Seite an dem Ereigniszeitpunkt berechnet, weshalb im zugehörigen Diagramm in Abbildung B.5 die Fehler von SI in der Nähe der Stelle  $t = 0.2$ s in der Größenordnung der Fehler des gesamten Zeitintervalls  $[t^0, t^\infty]$  bleiben.  $\square$

### Beispiel 5.5 (Sensitivitäten des verlustbehafteten Getriebes)

In Beispiel 3.22 wird die Lösung für das Getriebemodell mit Leistungsverlusten analytisch konstruiert. Die Zeitabschnitte, während derer sich das Getriebe nicht dreht, bedingen die Berechnung einer Filippov-Lösung. Wir integrieren das Modelica-Modell (Abbildung 3.4) mit FISEMO auf dem Zeitintervall  $[0, 0.5]$  mit unterschiedlichen Toleranzen  $TOL := RTOL_i = ATOL_i$  ( $i = 1, \dots, n_x$ ) und betrachten die numerischen Lösungen für die Winkelgeschwindigkeit  $\omega_1$  in Abbildung 5.3 (links).

Die Filippov-Lösungen werden für alle Toleranzen korrekt berechnet, eine sichtbare Abweichung ist im gesamten Lösungverlauf nur für  $TOL = 10^{-3}$  zu erkennen. Im rechten Diagramm zeigen die skalierten Fehler (analog zu den Ausdrücken (5.19) mit der analytischen Lösung als Referenzlösung), dass die gemessenen Genauigkeiten die vorgegebenen Fehlertoleranzen höchstens um den Faktor 10 überschreiten. In den Phasen der Filippov-Lösung stimmen die numerischen Werte unabhängig von den Toleranzvorgaben sehr genau mit der Lösung überein (Fehler  $< 10^{-10}$ ). Es ist bei diesem Beispiel nicht zu beobachten, dass sich die numerische Filippov-Lösung von der Schaltfläche entfernt (kein Drift-off).

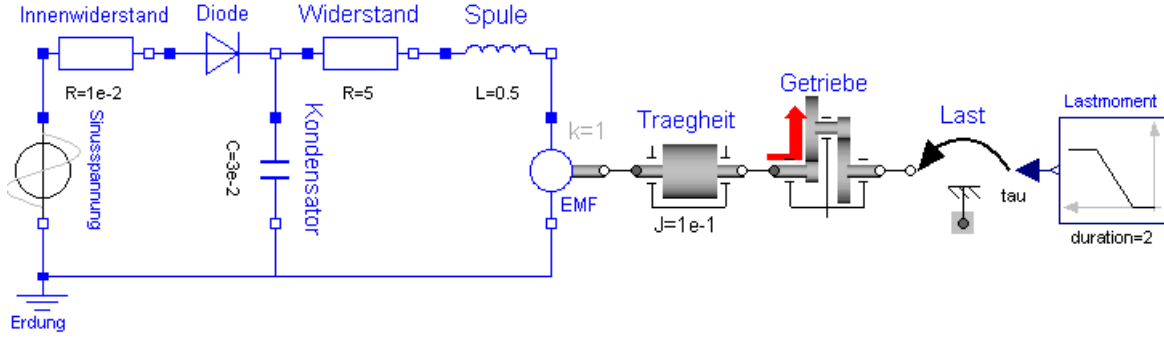


Abbildung 5.4: Modelica-Modell des belasteten Elektromotors mit Getriebe, vgl. auch Abbildung 2.2.

In einem zweiten Schritt werden für die Parameter  $p = (J_1, \eta)^T$  aus Tabelle 3.5 die Sensitivitäten von  $\omega_1$  wie in Beispiel 5.4 mit SI und FD ermittelt. Die Zeitverläufe der FD-Sensitivitäten zeigen im Vergleich zu den Referenzsensitivitäten für Parameter 1 optisch sehr gute Ergebnisse (siehe Abbildung B.1), die auch in den Fehlerdiagrammen von Abbildung B.2 bestätigt werden. Die Genauigkeiten unterschreiten die geschätzten theoretischen Fehler  $\sqrt{\text{TOL}_{\text{FD}}}$  um ein bis zwei Größenordnungen.

Für Parameter 2 ergibt sich ein etwas anderes Bild. Hier sind die Sensitivitäten vor jeder Phase mit Filippov-Lösung unstetig. Diese Unstetigkeiten werden – wie schon in Beispiel 5.4 – von den externen Differenzenquotienten nur angenähert. Folglich treten große Fehler in der Nähe der entsprechenden Ereigniszeitpunkte zutage. In den dynamischen Lösungsabschnitten entspricht die Genauigkeit der FD-Sensitivitäten bis auf einen Faktor 10 annähernd den Erwartungen. Während der Filippov-Phasen sind teilweise deutlich genauere Sensitivitätsergebnisse bzgl. beider Parameter sowohl für FD als auch für SI zu beobachten. Die SI-Fehler ähneln sich für beide Parameter auf dem gesamten Simulationszeitraum und überschreiten die gewählten Toleranzen maximal um das Zehnfache. Die Ergebnisse demonstrieren, dass die Filippov-Lösungen und Sensitivitäten für dieses Modelica-Modell von FISEMO numerisch zuverlässig berechnet werden.  $\square$

### Beispiel 5.6 (Belasteter Elektromotor mit Getriebe)

Als Fortführung von Beispiel 5.4 und 5.5 werden für das kombinierte, etwas komplexere Modelica-Modell aus Abbildung 5.4 analoge numerische Tests wie für Beispiel 5.4 durchgeführt. Das Modell enthält zwischen Motor und Belastungsvorgabe das Getriebe, das zeitweise Filippov-Lösungen induziert. Das Belastungsmoment  $M_L$  wird hierfür wie folgt festgelegt:

$$M_L(t) = \begin{cases} -50, & \text{falls } t < \tau, \\ -50 + \frac{\overline{M}}{2}(t - \tau), & \text{falls } \tau < t < \tau + 2, \\ -50 + \overline{M}, & \text{falls } \tau + 2 < t \end{cases}$$

mit den Werten  $\tau = 0.2 \text{ s}$  und  $\overline{M} = 8 \text{ Nm}$ . In Abbildung 5.5 unterteilt sich die numerische Lösung der lastseitigen Drehgeschwindigkeit  $\omega_2$  in 15 abwechselnde Phasen mit klassischer Lösung ( $\omega_2 \neq 0$ ) und Filippov-Lösung ( $\omega_2 = 0$ ). Ein ausreichendes Antriebsmoment für

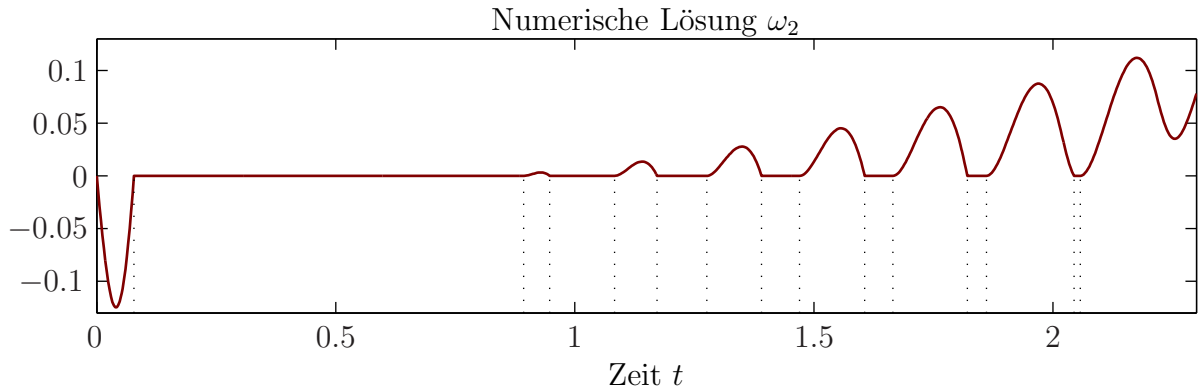


Abbildung 5.5: Numerische Lösung für Beispiel 5.6 (belasteter Elektromotor mit Getriebe), berechnet in FISEMO mit  $\text{TOL} = 10^{-9}$ .

eine durchgängige Drehbewegung ist erst nach einigen Oszillationen der Versorgungsspannung verfügbar.

Die Sensitivitäten  $(\omega_2)_p$  bzgl. der acht Parameter  $p = (R_i, R, C, L, k, J, \overline{M}, \tau)^T$  finden sich in den Zeitverläufen in Abbildung B.8 sowie den ausführlichen Fehlerdiagrammen der Abbildungen B.9 bis B.12 wieder. Es ist ersichtlich, dass die Sensitivitäten aller Parameter vor den Übergängen zu Filippov-Lösungen jeweils Unstetigkeiten besitzen. Entsprechend ungenaue Resultate liefert die externe numerische Differentiation in der Nähe der Ereignispunkte. Aber auch jenseits der Ereignispunkte sind die Genauigkeiten der FD-Sensitivitäten für grobe Toleranzen außerhalb der Filippov-Phasen z. T. deutlich unbefriedigend (Parameter 1, 2) und uneinheitlich.

Die Ergebnisse der Sensitivitätsintegration halten für  $\text{TOL}_{\text{SI}} \geq 10^{-7}$  fast alle (bis auf die Resultate von Parameter 2) die Fehlertoleranzen ein und unterstreichen damit die robuste numerische Lösung auch für größere Toleranzvorgaben. Die Anforderung  $\text{TOL}_{\text{SI}} = 10^{-8}$  kann insbesondere für die Parameter 2 und 5 nicht durchgängig erfüllt werden. Ähnliche Probleme sind bereits in Beispiel 5.4 aufgetreten. Ein möglicher Grund hierfür könnte die Abschätzung (5.10) sein, die die Genauigkeit der Sensitivitätszustände nur für

$$\text{TOL}_{\text{SI}} \geq \text{eps}_f^{2/3} = \text{eps}_M^{4/9} \doteq 10^{-7}$$

garantiert. Auch bei diesem Beispiel ist festzustellen, dass die Sensitivitäten in den Phasen der Filippov-Lösung für SI und FD nahezu unabhängig von den Fehlertoleranzen sehr genau berechnet werden.  $\square$

### Beispiel 5.7 (Drei Elektromotoren mit Getriebe)

Die Eigenschaft von FISEMO, Modelica-Modelle mit gleichzeitig aktiven Schaltfunktionen bzw. zeitlich parallelen Filippov-Lösungen (vgl. Bemerkung 3.24) integrieren zu können, zeigen wir an den folgenden Beispielen.

Wir instanzieren das Modell des Elektromotors mit Getriebe (Beispiel 5.6) dreimal in ein neues Modell. Damit werden drei identische Teilsysteme zu einem System zusammengefasst und als solches gelöst. Die Sensitivitätsberechnungen werden mit verschiedenen



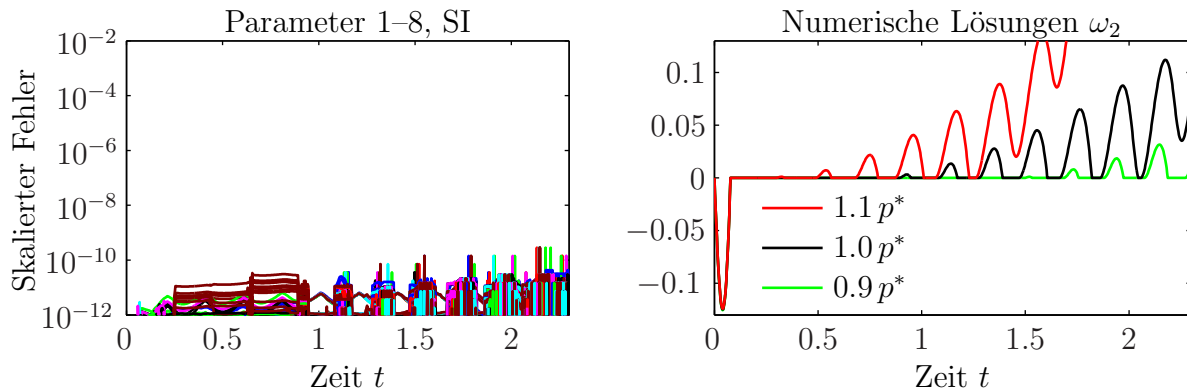


Abbildung 5.6: Links: Skalierte Differenz zwischen den Sensitivitäten des ersten Teilmodells und der beiden anderen identischen Teilmodelle aus Beispiel 5.7. Rechts: Numerische Lösungen  $\omega_2$  für die drei unterschiedlichen Teilmodelle aus Beispiel 5.7, berechnet in FISEMO mit  $\text{TOL} = 10^{-9}$ .

Fehlertoleranzen für die acht Parameter aus Beispiel 5.4 durchgeführt. In Abbildung 5.6 (links) bleiben die skalierten Differenzen zwischen den Sensitivitäten  $(\omega_2)_p$  des ersten Modells und der beiden zusätzlichen Modelle unter  $10^{-9}$ . Dies gilt sowohl für alle Parameter als auch für alle Toleranzvorgaben, womit insgesamt die Differenzen zwischen den Ergebnissen der Teilmodelle geringer sind als die zu erwartenden numerischen Fehler auf Grund der Fehlertoleranzen.

In einem zweiten Versuch weisen wir den drei Teilmodellen unterschiedliche Parameterwerte zu. Das erste Modell habe den Wert  $p^*$  aus Beispiel 5.6. Dem zweiten Modell werde der Wert  $1.1p^*$  und dem dritten die Größe  $0.9p^*$  übergeben. Die resultierenden Zeitverläufe für  $\omega_2$  sind in Abbildung 5.6 (rechts) dargestellt und geben einen Eindruck, wie bereits bei diesem noch überschaubaren Modell durchaus komplexere Vorgänge durch die zeitlich überlappenden, parallelen Filippov-Lösungen zustande kommen. Die entwickelte Software erkennt automatisch die einzelnen Phasen und berechnet die zugehörigen rechten Seiten für die verschiedenen Komponenten des Zustandsvektors.  $\square$

Die Ergebnisse dieses Abschnitts belegen, dass die entwickelte Software in der Lage ist, numerische Lösungen und Sensitivitäten von Modellen mit den in Bemerkung 5.1 angegebenen Eigenschaften korrekt zu berechnen. Weitere Bestandteile der Verifizierung sind die Fehleranalysen in den Beispielen 5.4 bis 5.6, die insbesondere an Ereigniszeitpunkten mit unstetigen Sensitivitäten durch die externen Differenzenquotienten bestätigt werden. Auf einen Vergleich der Rechenzeiten für die Zeitintegration der Sensitivitätsgleichungen bei verschiedenen Fehlertoleranzen wurde in diesem Kapitel verzichtet, da in Kapitel 6 ausführliche Rechenzeitvergleiche für ein Robotermodell im Rahmen einer Optimierung vorgenommen werden.



## 6 Anwendung: Gradientenbasierte Parameteridentifizierung eines Robotermodells

Eine wesentliche Anwendung der Sensitivitätsanalyse dynamischer Systeme besteht in der Parameteroptimierung derartiger Systeme. Effiziente numerische Optimierungsverfahren nutzen Ableitungsinformationen der Zielfunktionen, so dass sich hier ein Anknüpfungspunkt zwischen Sensitivitätsanalyse und Optimierung ergibt. Hinreichend genaue Sensitivitäten tragen mitunter zur Konvergenz der iterativen Optimierungsverfahren und zur Qualität des Optimierungsergebnisses bei. Abschnitt 6.1 führt in die Thematik der Parameteridentifizierung von Systemmodellen sowie der numerischen Optimierung für die zu Grunde liegende Zielfunktion ein. Numerische Tests an einem Modelica-Robotermodell liefern in Abschnitt 6.2 Ergebnisse von Optimierungsläufen. Zum einen wird dadurch die Funktionalität der Softwareumgebung FISEMO an einem komplexen Modell für Industrieanwendungen unter Beweis gestellt, zum anderen illustrieren die Ergebnisse die Vorteile der in der Arbeit gewählten Sensitivitätsanalyse im Vergleich zur externen numerischen Differentiation.

### 6.1 Parameteridentifizierung und numerische Lösungsansätze

Parameteridentifizierung beschäftigt sich mit der Berechnung optimaler Parameterwerte eines parameterabhängigen dynamischen Systems. Die Güte der Parameter  $p$  bewertet man hierbei mit einer zu minimierenden Zielfunktion  $C(p)$ , die den Unterschied zwischen Simulationsergebnissen und Messreihen darstellt. Im ersten Abschnitt definieren wir das Problem der Parameteridentifizierung. Anschließend beschreiben wir das SQP-Verfahren [Spe93, NW06] und die Optimierungsumgebung MOPS [Joo07] zur numerischen Lösung des Optimierungsproblems.

#### 6.1.1 Identifizierung von Modellparametern

Als wichtiger Bestandteil eines allgemeinen Modellbildungs- bzw. Systemidentifizierungsprozesses [Ise88a, Maj98, Sch04a] gleicht man in der Parameteridentifizierung ein parametrisiertes Modell mit fester Modellstruktur an Messdaten des beschriebenen Systems

an. Wir nehmen an, die physikalischen Größen der Ausgabegrößen  $y = g(x, p, t)$  eines parameterabhängigen Anfangswertproblems (5.1) werden zu diskreten Zeitpunkten  $t_i$  ( $i = 0, \dots, N$ ) gemessen und mit  $\gamma_{ji}$  ( $j = 1, \dots, n_y$ ) bezeichnet. Es gibt verschiedene Ansätze, den Unterschied zwischen Modell und Messung quantitativ zu definieren. Sehr verbreitet ist ein Modellgleich im Sinn der kleinsten Fehlerquadrate, also

$$C(p) := \sum_{j=1}^{n_y} \sum_{i=0}^N (y_j(p, t_i) - \gamma_{ji})^2$$

bzgl.  $p$  zu minimieren. Statistische Überlegungen [Boc87] beziehen Streuungen  $\sigma_{ji}$  der Messergebnisse in die Quadratsumme mit ein, was zu einer gewichteten Summe

$$C(p) := \sum_{j=1}^{n_y} \sum_{i=0}^N \frac{1}{\sigma_{ji}^2} (y_j(p, t_i) - \gamma_{ji})^2 \quad (6.1)$$

führt. Unter bestimmten Annahmen erhält man aus der Minimierung von  $C$  den Maximum-Likelihood-Schätzwert [Ise88b] für die Parameter  $p$ . Weiter verallgemeinert ergibt sich

$$C(p) := \sum_{j=1}^{n_y} \int_{t_0}^{t_N} \omega_j(t) (y_j(p, t) - \hat{y}_j(t))^2 dt. \quad (6.2)$$

Hierbei werden die diskreten Messwerte  $\gamma_{ji}$  durch zeitabhängige Funktionen  $\hat{y}_j$  ersetzt, die den Verlauf der Messwerte interpolieren oder approximieren. Weiter werden die Gewichte  $\sigma_{ji}^{-2}$  aus (6.1) zu zeitabhängigen Gewichtsfunktionen  $\omega_j$  verallgemeinert. Der Ansatz (6.2) wird als  $L_2$ -Kriterium zur Parameteridentifizierung bezeichnet [ND83, Boc87, Hei92]. Für interpolierende und zwischen den Zeitpunkten  $t_i$  stückweise konstante Funktionen  $\hat{y}_j$  ergibt sich mit geeigneten Gewichtsfunktionen genau die Darstellung von (6.1).

Grundsätzlich stellt sich die Frage, welche Parameter aus der Minimierung von  $C$  eindeutig rekonstruiert werden können. Beispielsweise können die Induktivitäten  $L_1, L_2$  von zwei in Serie geschalteten linearen elektrischen Spulen, vgl. auch Beispiel 2.1, zusammen nicht eindeutig aus Strommessungen bestimmt werden. Diese Tatsache ist darin begründet, dass nur die Summe  $L_1 + L_2$  in die Komponentengleichung  $v = (L_1 + L_2) \frac{di}{dt}$  und damit in die Zielfunktion  $C$  eingeht. Des Weiteren lassen sich Parameter, die kaum oder gar keinen Einfluss auf  $C$  haben, nur sehr ungenau bzw. gar nicht ermitteln. Diese Aussage gilt gleichermaßen für jedes Optimierungsproblem – auch jenseits von Parameteridentifizierungsaufgaben. Es existieren mehrere Methodiken, ein Maß für die Identifizierbarkeit von Parametern zu definieren. Alle in Bemerkung 6.1 vorgestellten Techniken basieren auf Linearisierungen bzgl. der Parameter und damit auf der Sensitivitätsanalyse.

### Bemerkung 6.1 (Identifizierbarkeitsmaße)

In einem ersten, pragmatischen Ansatz bewerten wir Parameteränderungen in der Nähe der Startparameter  $p^0$  mit Hilfe der relativen Sensitivitäten [Maj98]

$$b_i := \frac{p_i^0}{C(p^0)} C_{p_i}(p^0) \quad (p_i^0, C(p^0) \neq 0). \quad (6.3)$$

Mit dem in erster Näherung geltenden Zusammenhang

$$\frac{C(p^0 + \Delta p_i) - C(p^0)}{C(p^0)} \doteq b_i \frac{\Delta p_i}{p_i^0}$$

kennt man lokal die relative Auswirkung einer relativen Parameteränderung auf die Zielfunktion  $C$ . Die Werte  $b_i$  sind skalierungsinvariant gegenüber linearen Skalierungen der Parameter und der Zielfunktion. Je größer die Beträge der relativen Sensitivitäten sind, umso wichtiger ist es, die Werte der zugehörigen Parameter lokal um  $p^0$  zu verbessern. Kleine Betragswerte von  $b_i$  lassen darauf schließen, dass die entsprechenden Parameter lokal schlecht identifiziert werden können.

Weitere Möglichkeiten der Parameterklassifikation beruhen auf der Maximum-Likelihood-Schätzung. Die skalierte *Fisher-Informationsmatrix* [Maj98]

$$M = (m_{ij}) := H^T \Sigma_{\text{Mess}}^{-1} H \in \mathbb{R}^{n_p \times n_p}$$

ergibt sich mit der Matrix  $H := Y_p(p^0) \text{diag } p^0$  aus der Normalengleichung des linearisierten Parameteridentifizierungsproblems. Hierfür sei der Vektor

$$Y(p) := (y_1(p, t_0), \dots, y_1(p, t_N), \dots, y_{n_y}(p, t_0), \dots, y_{n_y}(p, t_N))^T$$

definiert. Die Kovarianzmatrix  $\Sigma_{\text{Mess}} \in \mathbb{R}^{n_Y \times n_Y}$  enthält die Varianzen der Messdaten für jede Komponente von  $y$  und jeden Zeitpunkt  $t_i$ . Es gelte die Voraussetzung, dass die Fehler der Messwerte unkorreliert und normalverteilt mit Erwartungswert Null seien. In [Sch04a] wird als Maß der Identifizierbarkeit des Parameters  $p_i$  unter anderem die Größe  $m_{ii}$  verwendet.

Alternativ kann auch die Kovarianzmatrix  $K = (k_{ij}) := M^{-1}$  der Parameterschätzung bzgl. der Messfehler betrachtet werden, falls  $M$  regulär ist. Je kleiner die relativen Parameterstreuungen  $\sqrt{k_{ii}}$  ausfallen, desto unempfindlicher reagieren die Parameterschätzungen in der Nähe von  $p^0$  auf Fehler der Messdaten. Eine singuläre Matrix  $M$  deutet auf nicht identifizierbare Parameter hin.

Majer [Maj98] kombiniert die Analyse der Identifizierbarkeit mit der Identifizierung und geht schrittweise vor. Die Parameter werden in dem Verfahren über die geschätzten Streuungen von transformierten Parameterwerten bewertet. Die transformierten Parameter werden dabei in der zur symmetrischen Matrix  $M$  gehörenden Eigenvektorbasis dargestellt. Der kleinste Eigenwert  $\lambda_{\min}$  von  $M$  trägt zur größten Streuung  $\sigma = \sqrt{1/\lambda_{\min}}$  der Parameter bei. Überschreitet  $\sigma$  eine vorgegebene Schranke, so wird die betragsgrößte Komponente des Eigenvektors zu  $\lambda_{\min}$  bestimmt. Der dazugehörige Parameter wird sodann aus dem Identifizierungsproblem entfernt, indem er konstant gehalten wird. Die damit in der Dimension reduzierte Informationsmatrix  $M$  kann dann analog untersucht werden. Diese Prozedur wird solange fortgeführt, bis die Streuung  $\sigma$  die Vorgabe unterschreitet. Nach der Identifizierung der ausgewählten Parameter wird eine erneute Analyse aller Parameterstreuungen ausgeführt, um so iterativ möglichst viele Parameter erfolgreich zu identifizieren.  $\square$

In [Boc87, Gru96] wird eine abschließende statistische Analyse der als optimal ermittelten Maximum-Likelihood-Schätzwerte  $p^*$  durchgeführt. Zu den Parameterwerten werden Konfidenzintervalle  $[p^* - z, p^* + z]$  berechnet, die zu einer vorgegebenen Wahrscheinlichkeit  $\beta \in (0, 1)$  den Bereich abschätzen, in dem sich die gesuchten Parameterwerte befinden. Die halben Intervalllängen  $z_i$  berechnen sich zu

$$z_i := p_i^* \sqrt{\chi_{\beta}^2(n_p) k_{ii}^*}. \quad (6.4)$$

In der Formel ist das  $\beta$ -Quantil der  $\chi^2$ -Verteilung mit  $n_p$  Freiheitsgraden enthalten sowie ein Diagonalelement der Kovarianzmatrix  $K^* = K(p^*)$ , die im ermittelten Optimalpunkt ausgewertet ist. Je kleiner die Intervalle sind, umso geeigneter sind die Betriebspunkte der Versuchsreihe, die unbekannten Parameter zu bestimmen. Eine zufriedenstellende Parameteridentifizierung ist erst dann abgeschlossen, wenn mit den berechneten Parametern  $p^*$  hinreichend genaue Modellaussagen auch für Betriebspunkte und Anregungen, die nicht direkt in die Bestimmung von  $p^*$  einfließen, möglich sind (Extrapolation). Voraussetzung hierfür ist eine korrekte Modellstruktur, die im Falle unbefriedigender Extrapolationsergebnisse korrigiert werden sollte [Ise88a, Maj98].

### 6.1.2 Numerische Lösung des Optimierungsproblems – SQP-Verfahren

Die Minimierung der Fehlerfunktion  $C$  aus Abschnitt 6.1.1 führt zu einem nichtlinearen Optimierungsproblem, das oft Beschränkungen in den Parametern besitzt:  $\pi \leq p \leq \Pi$ . Die Vektoren  $\pi, \Pi \in \mathbb{R}^{n_p}$  sind dabei feste Grenzen, zwischen denen die Parameterwerte variieren können. Beschränkte nichtlineare Optimierungsprobleme betrachtet man allgemein in der Form

$$\min_p C(p) \quad \text{unter} \quad \begin{cases} h_j(p) = 0, & j = 1, \dots, n_G, \\ h_j(p) \geq 0, & j = n_G + 1, \dots, n_h. \end{cases} \quad (6.5)$$

Der Parameterraum, in dem das Minimum der Zielfunktion  $C : \mathbb{R}^{n_p} \rightarrow \mathbb{R}$  gesucht werden soll, wird durch Gleichheits- und Ungleichheitsbedingungen mit den Komponenten der Funktion  $h : \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n_h}$  eingeschränkt. Mit Hilfe der Lagrange-Funktion

$$L(p, v) := C(p) - \sum_{j=1}^{n_h} v_j h_j(p), \quad v_j \in \mathbb{R},$$

lassen sich beschränkte Optimierungsprobleme umformulieren. Beispielsweise existieren für eine lokale Optimalstelle  $p^*$  von (6.5) unter gewissen Voraussetzungen an  $C$  und  $h$  Lagrange-Multiplikatoren  $v_j^* \in \mathbb{R}$  ( $j = 1, \dots, n_G$ ) und  $v_j^* \geq 0$  ( $j = n_G + 1, \dots, n_h$ ), mit denen die Identität  $\nabla L(p^*, v^*) = 0$  gilt [Spe93].

Numerische Verfahren zur Lösung von (6.5) erzeugen ausgehend von einem Startvektor  $p^0$  iterativ verbesserte Lösungen  $p^k$  ( $k \geq 0$ ), bis der Algorithmus von einem Konvergenz- oder Abbruchkriterium beendet wird. In der Literatur [Spe93, NW06] wird das SQP-Verfahren

als effiziente universell einsetzbare Optimierungsmethode angesehen. Die Idee der sequentiellen quadratischen Optimierung (englisch: Sequential Quadratic Programming, SQP) beruht auf der sukzessiven Approximation von (6.5) durch eine quadratische Zielfunktion mit linearisierten Nebenbedingungen [Sch81]:

$$\min_d d^\top B^k d + C_p(p^k)d \quad \text{unter} \quad \begin{cases} h'_j(p^k)d + h_j(p^k) = 0, & j = 1, \dots, n_G, \\ h'_j(p^k)d + h_j(p^k) \geq 0, & j = n_G + 1, \dots, n_h. \end{cases} \quad (6.6)$$

Die Matrix  $B^k \in \mathbb{R}^{n_p \times n_p}$  ist dabei eine positiv definite Approximation der Hessematrix  $L_{pp}$  im Optimalpunkt. Das quadratische Ersatzproblem (6.6) wird mit Methoden der quadratischen Optimierung gelöst, auf die wir nicht näher eingehen wollen. Details finden sich etwa in [Spe93, NW06]. Die optimale Lösung von (6.6) sei mit  $d^k$  und die zugehörigen Lagrange-Multiplikatoren seien mit  $u^k$  bezeichnet. Daraus berechnen sich mit einer Schrittweite  $\alpha_k > 0$  die neuen Iterierten  $p^{k+1} := p^k + \alpha_k d^k$  und  $v^{k+1} := v^k + \alpha_k(u^k - v^k)$ . Als Startwert kann  $v^0 := 0$  gesetzt werden. Um  $\alpha_k$  zu bestimmen, wird die Testfunktion

$$\Phi(\alpha) := \tilde{\Phi}((p^k, v^k) + \alpha(d^k, u^k - v^k))$$

mit der erweiterten Lagrange-Funktion  $\tilde{\Phi} : \mathbb{R}^{n_p+n_h} \rightarrow \mathbb{R}$  minimiert [Sch81]. Bevor der Iterationszähler  $k$  um Eins erhöht wird, berechnet man die Matrix  $B^{k+1}$  aus der BFGS-Formel

$$B^{k+1} := B^k + \frac{1}{a^k{}^\top \Delta p^k} a^k a^k{}^\top - \frac{1}{\Delta p^k{}^\top B^k \Delta p^k} B^k \Delta p^k \Delta p^k{}^\top B^k$$

mit

$$\Delta p^k := p^{k+1} - p^k, \quad a^k := \theta \bar{a}^k + (1 - \theta) B^k \Delta p^k \quad \text{und} \quad \bar{a}^k := L_p^\top(p^{k+1}, u^k) - L_p^\top(p^k, u^k).$$

Wenn  $B^k$  positiv definit ist, dann kann der Wert  $\theta \in [0, 1]$  stets so gewählt werden, dass  $B^{k+1}$  ebenfalls positiv definit wird. Für den Beginn der Iteration muss  $B^0$  eine positiv definite Matrix, z. B. die Einheitsmatrix, sein.

Aus software-technischer Sicht benötigt man neben dem Optimierungsalgorithmus auch Schnittstellen zwischen der Optimierungsroutine und der Auswertung von  $C$ ,  $h$  sowie den Ableitungen  $C_p$ ,  $h_p$ . Für die numerischen Untersuchungen im nächsten Abschnitt verwenden wir die *Optimierungsumgebung* MOPS (englisch: Multi-Objective Parameter Synthesis) [Joo07], die eine einheitliche Oberfläche zur Definition und Lösung von Optimierungsproblemen sowie zur Visualisierung des Optimierungsfortschritts und der Ergebnisse bietet. Grundstock der Umgebung bildet eine Auswahl an verschiedenen numerischen Optimierungsalgorithmen, die alle auf die einmal zu definierende Problemformulierung zugreifen können. Neben ableitungsfreien Methoden wie Simplex-Verfahren oder genetischen Algorithmen enthält MOPS Verfahren, die auf Gradienten zurückgreifen.

Die SQP-Implementierung in MOPS ist eine Variante aus der RASP-Bibliothek [GJ91], die insbesondere für regelungstechnische Entwurfsprobleme eingesetzt wurde. Zusätzlich zu den regelungstechnischen Aufgaben [JBL<sup>+</sup>02] werden mit MOPS beispielsweise verbesserte Flugzeugsystemarchitekturen [SPB06] gesucht. Besonderes Augenmerk wird in der Optimierungsumgebung darauf gelegt, mehrere Entwurfskriterien (Zielfunktionen) gleichzeitig

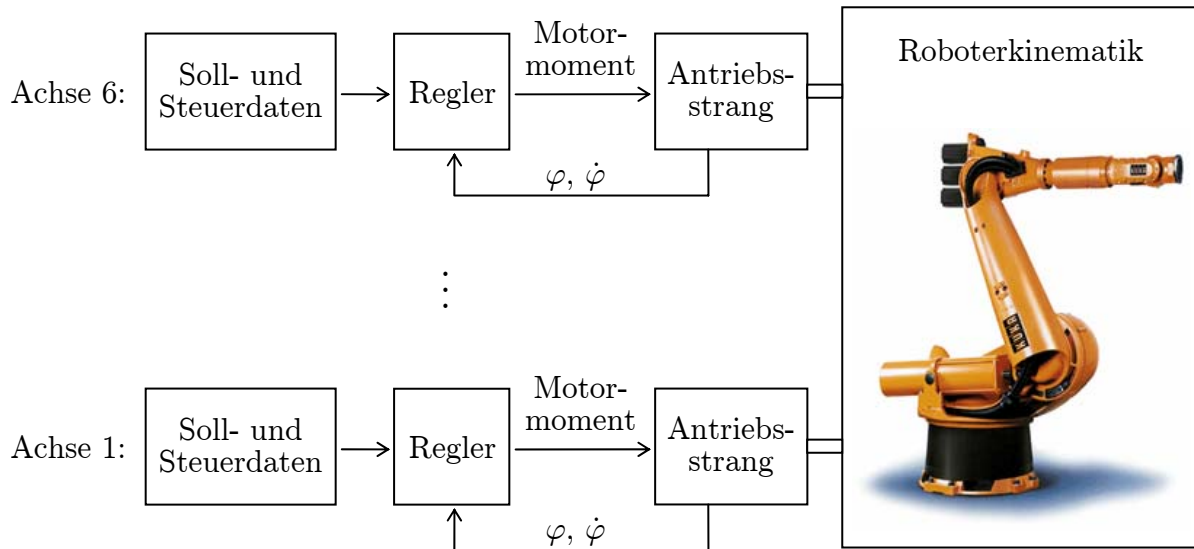


Abbildung 6.1: Prinzipieller Aufbau des Modelica-Modells für einen KUKA-Industrieroboter.

formulieren und mit verschiedenen Ansätzen auf eine Zielfunktion für den Optimierungsalgorithmus reduzieren zu können. Als grafisches Äquivalent der mehrzieligen Problemformulierung sind die *parallelen Koordinaten* zu sehen, die die Kriteriensituation für einen konkreten Parametervektor visuell wiedergeben.

## 6.2 Numerische Ergebnisse für ein Robotermodell

Die Anforderungen der Industrierobotik wachsen kontinuierlich in Bezug auf die Bahngenauigkeit hochdynamischer Bewegungen. Einen wesentlichen Beitrag zur Bewältigung der anspruchsvollen Vorgaben leisten modellbasierte Regelalgorithmen. Ein dafür nötiges, komplexes Modell der Roboterdynamik wird im Folgenden genauer beschrieben und anschließend mit den Methoden der Parameteridentifizierung an Messergebnisse angeglichen. Numerische Tests beleuchten dabei speziell das Iterationsverhalten des SQP-Verfahrens sowie den Zusammenhang zu Sensitivitätsberechnungen.

### 6.2.1 Dynamisches Robotermodell in Modelica

Wir beschreiben das Modell<sup>1</sup> eines sechssachsigen Knickarmroboters der Firma KUKA<sup>2</sup>, das in [Thü06] ausführlich besprochen wird. Da die Modellierung in Modelica erfolgt, richtet sich der Modellaufbau in Abbildung 6.1 nach der physikalischen Struktur des Roboters. Der Roboter besteht aus oft starr angenommenen Gliedern und elastischen Gelenken.

<sup>1</sup>Der Autor ist M. Reiner (DLR Oberpfaffenhofen) dankbar für die Bereitstellung des Modelica-Modells und der Messdaten.

<sup>2</sup>KUKA Roboter GmbH, [www.kuka.com](http://www.kuka.com).



Diese kinematisch offene Kette wird mit entsprechend parametrisierten Teilmodellen für die Strukturelemente und Gelenke realisiert, die komponentenbezogene Gleichungen der dreidimensionalen Mehrkörperdynamik enthalten. Die Basisklassen für die Teilmodelle sind der Modelica-Standardbibliothek `Modelica.Mechanical.Multibody` entnommen.

Neben der Mehrkörperdynamik spielen (eindimensionale) Effekte in den Antriebssträngen eine wesentliche Rolle. In einem Antriebsstrang sind der Motor, das Getriebe und die Leistungselektronik zusammengefasst. Elektrische Bestandteile des Antriebsstrangs werden in dem vorliegenden Modell vernachlässigt, da das elektrische Verhalten wesentlich schneller als das mechanische ist. Jedes Antriebsstrangmodell beschreibt neben der Motorträgheit die Übersetzung, Elastizität und Dämpfung des Getriebes sowie die Reibung in der Motorlagerung, im Getriebe und im Gelenk. Die Elastizitäten in den Getrieben werden dabei von nichtlinearen Kennlinien modelliert, um die Dynamik der Antriebsstränge möglichst realitätsnah wiederzugeben. Das Reibungsmodell kombiniert die verschiedenen Quellen von Reibungseinflüssen und bildet Coulomb-Reibung mit viskoser Reibung ab. Das Reibmoment  $M_R$  berechnet sich mit dem Gelenkwinkel  $\varphi$  und positiven Parametern  $R_v$ ,  $R_h$  geschwindigkeitsabhängig zu

$$M_R(\dot{\varphi}) = \begin{cases} R_v \dot{\varphi} + R_h, & \text{falls } \dot{\varphi} > 0, \\ R_v \dot{\varphi} - R_h, & \text{falls } \dot{\varphi} < 0. \end{cases}$$

Dieses Reibungsmodell kann zu inkonsistentem Schalten (siehe Abschnitt 3.1.2) führen und Filippov-Lösungen bedingen. Da Reibung in allen Achsen des Roboters modelliert ist, treten Filippov-Lösungen parallel in Teilmodellen auf und müssen mit den Ansätzen aus den Abschnitten 3.2.2 bzw. 4.2 untersucht werden. Hier ergeben sich aus methodischer Sicht anspruchsvolle Strukturen, die denen aus Beispiel 5.7 ähneln, aber durch die Roboterkinematik verkoppelt sind. Die hier verwendete Technik, die Filippov-Lösung der unstetigen Reibungsmodelle *außerhalb* des Modells zu berechnen, ist in der vorliegenden Arbeit erstmals an Modelica-Modellen realisiert. Die bisher bekannte Simulation setzt Zustandsautomaten innerhalb jedes Reibungsmodells ein, um die Gleit- und Haftphasen zu unterscheiden [OEM99]. Ohne Unstetigkeiten kommen geglättete Modelle aus, deren Genauigkeit und numerische Eigenschaften dann aber zusätzlich von Glättungsparametern abhängen [Thü06].

Jede Achse des Roboters wird separat mit einem PPI-Kaskadenregler auf die motorseitige Position und Geschwindigkeit geregelt. Hierzu berechnet der Regler aus den Regelfehlern von Position und Geschwindigkeit zusammen mit einem Vorsteuermoment das aktuelle Motormoment, das der Elektromotor intern umsetzen muss. Wir beschränken uns für die numerischen Untersuchungen auf ein Modell mit Reglern der ersten drei Hauptachsen. Für die anderen Achsen wird im Folgenden das Reglermoment direkt aus Messungen vorgegeben. Insgesamt liegt ein nichtlineares, gekoppeltes dynamisches Modell mittlerer Dimension (40 Zustände, über 1100 nichttriviale Gleichungen) vor, das die Dynamik eines Industrieroboters inklusive der Regler abbildet. Es ist Grundlage für die Auslegung von verbesserten Reglern und Reglerstrukturen. Durch dynamische Invertierung von Modellvarianten lässt sich beispielsweise eine qualitativ hochwertige Vorsteuerung konstruieren [LTK<sup>+</sup>05]. Voraussetzung dafür ist eine gute Übereinstimmung der Simulationsergebnisse des Modells mit vergleichbaren Messungen am Roboterprüfstand.

### 6.2.2 Optimierungs- und Identifizierungsergebnisse

Einen wesentlichen Einfluss auf das Verhalten des Robotermodells haben Steifigkeits- und Dämpfungsparameter  $\tilde{p}$ . Da diese im Gegensatz zu Massen- und Geometrieparametern schwierig zu messen sind, werden sie mit dem Ansatz der Parameteridentifizierung aus Abschnitt 6.1.1 bestimmt. Werte für Dämpfungs- und insbesondere Steifigkeitskoeffizienten sind oftmals sehr groß ( $> 10^5$ ), und ihre möglichen Gültigkeitsbereiche können mehrere Größenordnungen umfassen. Daher hat es sich für das Robotermodell als vorteilhaft erwiesen, statt der Parameterwerte  $\tilde{p}$  die logarithmierten Größen  $p := \log_{10} \tilde{p}$  zu identifizieren.

Für die Bewertung der Parameter verwenden wir den Vergleich der kartesischen Beschleunigungen  $y := (a_1, a_2, a_3)$  am Roboterwerkzeug und formulieren Gleichung (6.2) wie folgt:

$$\dot{C}(p, t) = \sum_{j=1}^3 (a_j(p, t) - \hat{a}_j(t))^2, \quad C(p, t_0) = 0.$$

In dieser differenzierten Form lässt sich  $C$  direkt im Modelica-Modell definieren und damit auch  $C_p$  mit FISEMO berechnen. Als Näherungsfunktionen  $\hat{a}_j$  der Messdaten werden lineare Interpolationen tiefpassgefilterter Datenpunkte [Rei07] verwendet. Die Gewichtsfunktionen setzen wir  $\omega_j = 1$ , weil die Beschleunigungen in den drei Raumrichtungen gleichberechtigt und mit Messfehlern derselben Größenordnung behaftet sind. Daher nehmen wir an, dass die Kovarianzmatrix der Messwerte die Gestalt  $\Sigma_{\text{Mess}} = \sigma_{\text{Mess}}^2 I$  hat, die Messfehler also immer dieselbe Verteilung besitzen. In den linken Diagrammen von Abbildung C.1 in Anhang C sind die Zeitverläufe der gemessenen und der mit den Startparametern  $p^0$  simulierten Beschleunigungen dargestellt.

#### Bemerkung 6.2 (Auswahl der zu identifizierenden Parameter)

Zunächst untersuchen wir das Modell mit den in Bemerkung 6.1 vorgestellten Möglichkeiten, um die Parameter in den Startwerten nach ihrem Identifizierbarkeitsmaß zu sortieren. Da für die Reihenfolge der Parameter der Wert von  $\sigma_{\text{Mess}}$  irrelevant ist, setzen wir vorerst  $\sigma_{\text{Mess}} = 1$ . Tabelle 6.1 listet die Ergebnisse der vier Ansätze für 19 Parameter des Robotermodells auf. Die Berechnung der Sensitivitäten  $C_p$  und  $Y_p$  erfolgt mit FISEMO und einer Fehlertoleranz von  $\text{TOL}_{\text{SI}} = 10^{-7}$ . Die Parameter sind nach den Absolutwerten der relativen Sensitivitäten  $b_i$  sortiert. Die restlichen Kriterien sind nicht durchgängig monoton bzgl.  $i$ , kommen aber in der Tendenz zu ähnlichen Aussagen. Die markierten Einträge kennzeichnen in jeder Zeile der Tabelle die fünf Parameter, die dem Maß entsprechend am besten identifiziert werden können. Die ersten vier Parameter werden von jedem Kriterium in dieser Gruppe erkannt. Da sich auch die Werte der Maße für  $i > 4$  im Wesentlichen deutlich von denen für  $i \leq 4$  unterscheiden, wählen wir die Parameter  $p_1, \dots, p_4$  als die zu identifizierenden Parameter (zur Vereinfachung ebenfalls im Vektor  $p$  zusammengefasst) für die nachfolgenden Testrechnungen mit der SQP-Implementierung von MOPS aus.  $\square$

Neben der logarithmischen Skalierung der Parameter, die im Robotermodell selbst realisiert ist, wird zwischen dem eigentlichen Optimierungsalgorithmus und der numerischen



Tabelle 6.1: Identifizierbarkeitsmaße und Streuungen gemäß Bemerkung 6.1 am Beispiel des Robotermodells. Berechnung für die Startwerte  $p^0$ . Sortierung der Parameter nach  $|b_i|$ . **Farbliche** Kennzeichnung der fünf Parameter, die nach dem jeweiligen Kriterium am besten identifiziert werden können.

$i$	1	2	3	4	5	6	7	8	9	10
$ b_i $	<b>3.0</b>	<b>0.9</b>	<b>0.41</b>	<b>0.37</b>	<b>0.05</b>	0.04	0.02	0.012	0.010	0.008
$m_{ii}$	<b><math>10^6</math></b>	<b><math>4 \cdot 10^4</math></b>	<b><math>6 \cdot 10^4</math></b>	<b><math>5 \cdot 10^5</math></b>	30	266	1.8	71	0.9	0.4
$\sqrt{k_{ii}}$	<b>0.0001</b>	<b>0.003</b>	<b>0.008</b>	<b>0.009</b>	0.1	<b>0.04</b>	18	11	6.6	188
$\lambda_{\min}^{-1/2}$	<b>0.001</b>	<b>0.03</b>	<b>0.01</b>	<b>0.0015</b>	0.25	0.18	1.1	0.8	2.6	4.3

$i$	11	12	13	14	15	16	17	18	19
$ b_i $	0.007	0.005	0.004	0.0008	0.0004	$9 \cdot 10^{-5}$	$7 \cdot 10^{-5}$	$5 \cdot 10^{-5}$	$4 \cdot 10^{-5}$
$m_{ii}$	<b><math>5 \cdot 10^4</math></b>	1.9	0.1	0.02	0.03	0.008	0.0003	0.01	0.0001
$\sqrt{k_{ii}}$	0.2	57	82	982	222	297	$2 \cdot 10^4$	67	$6 \cdot 10^4$
$\lambda_{\min}^{-1/2}$	<b>0.16</b>	8.0	9.5	20	16	17	137	8.3	253

Integration der Modellgleichungen eine affine Skalierung der Form  $\bar{p} := Dp + d$  vorgenommen. Die Diagonalmatrix  $D$  und der Vektor  $d$  sind dabei so bestimmt, dass die vorgegebenen Unter- und Oberschranken  $\pi, \Pi$  für die Parameter  $p$  auf die Werte  $\pm 1$  abgebildet werden:  $\bar{p} \in [-1, 1]^{n_p}$ . Zu den Intervallvorgaben kommen keine weiteren Nebenbedingungen hinzu. Ausgehend von den Startparametern  $\bar{p}^0 = (0.65, -0.12, 0.16, 0.64)^T$  werden Serien von Optimierungsläufen durchgeführt, die sich in den Toleranzvorgaben für die Integration des Modells unterscheiden. Die Toleranzwerte  $\text{TOL}_{\text{Nom}} = \text{RTOL}_i = \text{ATOL}_i$  für die nominelle Integration des Modells allein (Auswertung von  $C$ ) und  $\text{TOL}_{\text{SI}} = \text{RTOL}_i = \text{ATOL}_i = \overline{\text{RTOL}}_{ij} = \overline{\text{ATOL}}_{ij}$  ( $i = 1, \dots, n_x, j = 1, \dots, n_p$ ) für die Integration des Modells mit den Sensitivitätsgleichungen (Auswertung von  $C_p$ ) variieren hierfür im Bereich zwischen  $10^{-2}$  und  $10^{-7}$ .

Die Abbruchschranke des SQP-Verfahrens wird für alle Tests konstant auf  $10^{-3}$  gesetzt, um die Optimierungsläufe vergleichen zu können. Die wesentlichen Ergebnisse können aus Tabelle C.1 entnommen werden. Neben der Anzahl der Auswertungen von Zielfunktion ( $\#C$ ) und Sensitivitäten ( $\#C_p$ ) sind die berechneten optimalen Parameter  $\bar{p}^*$ , der zugehörige Zielfunktionswert sowie die benötigte Rechenzeit aufgelistet. Die Anzahl der gelösten quadratischen Hilfsprobleme im SQP-Algorithmus entspricht der Anzahl der Auswertungen von  $C_p$ . Alle Berechnungen werden auf einem Laptop mit  $2 \times 2.2$  GHz CPU ausgeführt.

Das Ziel der Tests ist es, die Abhängigkeit des Optimierungsverlaufs und der Optimierungsergebnisse von den gewählten Toleranzvorgaben  $\text{TOL}_{\text{Nom}}$  und  $\text{TOL}_{\text{SI}}$  zu analysieren. In einem ersten Schritt betrachten wir dazu die Zielfunktionswerte  $C(p^*)$  aller Optimierungsläufe in den jeweiligen Optimalwerten  $p^*$ . In Abbildung 6.2 ist die Differenz  $C(p^*) - C^{\text{Ref}}$  in Abhängigkeit von  $\text{TOL}_{\text{SI}}$  und  $\text{TOL}_{\text{Nom}}$  dargestellt. Die Referenz-

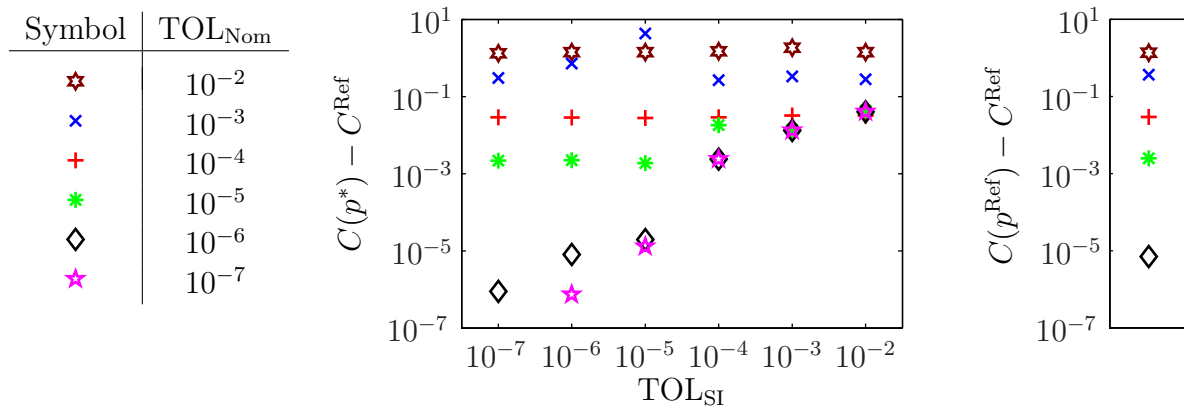


Abbildung 6.2: Links: Legende für die unterschiedliche Kennzeichnung der Ergebnisse mit  $TOL_{Nom}$  im Bereich zwischen  $10^{-2}$  und  $10^{-7}$ . Mitte: Abstand des berechneten optimalen Zielfunktionswerts vom Referenzwert  $C^{Ref}$  für alle Kombinationen der Toleranzvorgaben  $TOL_{SI}$  und  $TOL_{Nom}$ . Rechts: Numerische Integrationsfehler für  $p = p^{Ref}$  und verschiedene Fehlertoleranzen  $TOL_{Nom}$ .

größe  $C^{Ref} = 0.9904399187$  für den optimalen Zielfunktionswert resultiert dabei aus einer Optimierung mit den Einstellungen  $TOL_{Nom} = TOL_{SI} = 10^{-7}$ . Der zugehörige Parameterwert sei mit  $p^{Ref}$  bzw. in der skalierten Version mit  $\bar{p}^{Ref}$  bezeichnet. Die ermittelte Genauigkeit in der Zielfunktion hängt insbesondere von  $TOL_{Nom}$  ab. Die Zielfunktionswerte sind für  $TOL_{SI} \leq 10^{-6}$  in etwa so genau wie in der Integration des Modells mit dem konstanten Parameterwert  $p = p^{Ref}$  und verschiedenen Fehlertoleranzen  $TOL_{Nom}$  (siehe Abbildung 6.2 rechts). Die Vorgabe von  $TOL_{SI}$  im Bereich  $10^{-5}, \dots, 10^{-2}$  bewirkt eine Beschränkung der minimalen Fehler nach unten – insbesondere für wachsendes  $TOL_{SI}$ . Damit beeinflusst also neben  $TOL_{Nom}$  auch  $TOL_{SI}$  die Genauigkeit des Optimierungsergebnisses. Für die Werte  $TOL_{Nom} = 10^{-2}, 10^{-3}, 10^{-4}$  zeigen die Größenordnungen der Zielfunktionswerte im Optimum keine wesentliche Abhängigkeit von  $TOL_{SI}$ .

Der zweite Schritt der Analysen beleuchtet den rechenzeitabhängigen Verlauf der Zielfunktionswerte während der Optimierungen. Dazu beobachten wir in den Abbildungen C.2, C.3 und C.4 den Abstand  $C(p) - C^{Ref}$  für die Parameterwerte  $p$  der SQP-Iterationen. Jeder Wert von  $TOL_{SI}$  führt zu einer Gruppe von Optimierungsläufen mit verschiedenen Größen für  $TOL_{Nom}$ , die farblich gemäß Abbildung 6.2 unterschieden werden. Zu jedem  $TOL_{SI}$  zeigen Diagramme das Verhalten der Zielfunktionswerte in den ersten 60 Minuten Rechenzeit sowie die Größenordnung der Fehler insbesondere im Konvergenzbereich.

Es wird deutlich, dass grobe Fehlertoleranzen ( $\geq 10^{-4}$ ) von  $TOL_{Nom}$  und  $TOL_{SI}$  ausgehend von den Startparametern einen schnellen Abfall in der Zielfunktion bringen. Vergleicht man die Rechenzeiten für  $TOL_{Nom} = 10^{-4}, 10^{-5}, 10^{-6}$ , so ergeben sich in den Gruppen  $TOL_{SI} = 10^{-2}, 10^{-3}, 10^{-5}, 10^{-6}$  jeweils etwa gleiche Zeitspannen. Die Genauigkeiten, mit der die Zielfunktionen innerhalb einer Gruppe an die Referenzlösung heranreichen, unterscheiden sich jedoch für  $TOL_{SI} = 10^{-5}, 10^{-6}$ . Folglich erhält man in diesen Gruppen mit  $TOL_{Nom} = 10^{-6}$  bei gleichem Aufwand genauere Ergebnisse in der Zielfunktion als mit  $TOL_{Nom} > 10^{-6}$ . In den Gruppen  $TOL_{SI} \leq 10^{-5}$  bleibt die Differenz der Zielfunktion

vom Referenzwert für die meisten Werte von  $\text{TOL}_{\text{Nom}}$  für einige Iterationsschritte – und damit für eine gewisse Rechenzeit – auf dem hohen Niveau 4.3 nahezu konstant. Das SQP-Verfahren schafft es dennoch in allen Fällen mit  $\text{TOL}_{\text{Nom}} \leq 10^{-4}$ , den Zielfunktionswert auf die Referenzlösung zu reduzieren.

Die Parameteridentifizierung setzt ihren Schwerpunkt auf die Bestimmung der Parameter. Die Zielfunktion übernimmt dabei die Rolle einer Hilfsfunktion, um die optimalen Parameter beschreiben und berechnen zu können. Daher ist der letzte Schritt unserer Analysen die Beurteilung der Güte der Parameterwerte, die aus den 36 Optimierungsläufen in Tabelle C.1 hervorgehen. Die Diagramme in Abbildung C.5 sind nach den Werten für  $\text{TOL}_{\text{SI}}$  sortiert und visualisieren die optimalen Parameterwerte  $\bar{p}^*$ . Als Vergleich zu den Parameterschwankungen, die unterschiedliche Fehlertoleranzen des Integrators bewirken, sind die 95%-Konfidenzintervalle zu  $\bar{p}^{\text{Ref}}$  mit der beispielhaften Messfehlerstreuung  $\sigma_{\text{Mess}} = 0.5$  eingetragen. Die Intervalle werden gemäß Gleichung (6.4) aus der Sensitivitätsmatrix  $Y_p$  mit  $\text{TOL}_{\text{SI}} = 10^{-7}$  berechnet:  $\bar{z} = (0.029, 0.189, 0.046, 0.082)^T$ . Das Intervall um  $\bar{p}_2^{\text{Ref}}$  ist deutlich größer als die Intervalle der restlichen Parameter. Der Umstand, dass  $p_2$  folglich die Zielfunktion in der Nähe des Optimums weniger beeinflusst als die anderen Parameter, tritt auch in den Streuungen der numerischen, optimalen Parameterwerte für verschiedene Vorgaben von  $\text{TOL}_{\text{SI}}$  und  $\text{TOL}_{\text{Nom}}$  zutage.

Für jede Vorgabe von  $\text{TOL}_{\text{SI}}$  konvergieren die Parameter, wenn  $\text{TOL}_{\text{Nom}}$  gegen Null strebt. In den Fällen  $\text{TOL}_{\text{SI}} \leq 10^{-5}$ ,  $\text{TOL}_{\text{Nom}} \leq 10^{-4}$  sind die Parameterwerte in den Diagrammen nicht mehr zu unterscheiden. Für  $\text{TOL}_{\text{SI}} \geq 10^{-4}$  konvergieren die Parameter für  $\text{TOL}_{\text{Nom}} \rightarrow 0$  teilweise gegen „falsche“ Werte, die einen beträchtlichen Abstand zu den Referenzwerten besitzen. Bemerkenswert ist die Beobachtung, dass für  $\text{TOL}_{\text{SI}} = 10^{-7}$  sogar die von  $\text{TOL}_{\text{SI}}$  stark abweichenden Fehlertoleranzen  $\text{TOL}_{\text{Nom}} = 10^{-2}$ ,  $10^{-3}$  der Nominalintegration gute Identifizierungsergebnisse im Rahmen der Konfidenzintervalle liefern.

Insgesamt garantieren stringendere Fehlertoleranzen  $\text{TOL}_{\text{SI}}$  und  $\text{TOL}_{\text{Nom}}$  in der Parameteridentifizierung des Robotermodells sowohl Konvergenz gegen den optimalen Zielfunktionswert als auch Konvergenz der Parameterwerte gegen die Referenzparameter. Die Rechenzeiten verlängern sich entsprechend, insbesondere führt  $\text{TOL}_{\text{Nom}} = 10^{-7}$  im Vergleich zu  $\text{TOL}_{\text{Nom}} = 10^{-6}$  in allen Gruppen zu erheblichem Mehraufwand. Größere Werte ( $\geq 10^{-3}$ ) für  $\text{TOL}_{\text{Nom}}$  und  $\text{TOL}_{\text{SI}}$  bedingen in den untersuchten Fällen unbefriedigende Resultate in den optimalen Zielfunktions- und Parameterwerten. Um die Güte der Referenzparameter  $p^{\text{Ref}}$  zu testen, sind in Abbildung C.7 die Simulationsergebnisse des Robotermodells für andere Anregungsfunktionen und damit für andere Sollgrößen für Motorposition, Motorgeschwindigkeit und Vorsteuermoment vergleichend für die Parameter  $p = p^0$  und  $p = p^{\text{Ref}}$  dargestellt. Die simulierten Beschleunigungstrajektorien decken sich für  $p = p^{\text{Ref}}$  recht gut mit den gemessenen Größen, obwohl die zugehörigen Anregungsgrößen nicht in die Identifizierung von  $p$  eingegangen sind.

Die Ergebnisse der Parameteridentifizierung erwecken den Eindruck, dass es zu Anfang eines Optimierungslaufs ausreicht, grobe Toleranzvorgaben an die Integration zu stellen. Erst für die Konvergenz der Parameterwerte, wenn die Zielfunktion schon in der Größenordnung des Referenzwertes liegt, sind anscheinend kleinere Fehlertoleranzen notwendig. Aus mathematischer Sicht wäre es denkbar, ausgehend von größeren Werten die

Toleranzvorgaben für  $\text{TOL}_{\text{Nom}}$  und  $\text{TOL}_{\text{SI}}$  im Laufe der SQP-Iterationen automatisch zu verkleinern, um damit Rechenzeit im Vergleich zu konstant kleinen Werten einzusparen. In der Praxis bedeutet dieser Ansatz jedoch eine Kopplung des SQP-Verfahrens mit der Integration der Modellgleichungen, die nicht ohne Weiteres realisiert werden kann. Zudem stellt sich die (schwierige) Frage, wann auf welche Fehlertoleranzen umgeschaltet werden soll, damit man eine möglichst effizient – aber zuverlässig – berechnete Lösung erhält. Zuverlässigkeit ist in der praktischen Anwendung oftmals wichtiger als die Rechenzeit, solange sie sich in gewissen Grenzen hält.

Abschließend führen wir die Identifizierung der Parameter mit externer numerischer Differentiation (finite Differenzen) als Methode der Sensitivitätsberechnung durch. Unter denselben Rahmenbedingungen wie oben werden die Toleranzwerte  $\text{TOL}_{\text{Nom}}$  variiert. Die Gradienten der Zielfunktion werden mit einseitigen Differenzenquotienten angenähert. Als Parameterinkremente  $\Delta p_i$  der finiten Differenzen dienen – wie in Abschnitt 5.3 – die Größen  $p_i \sqrt{\text{TOL}_{\text{Nom}}}$ . Tabelle C.2 und Abbildung C.6 fassen die Resultate in derselben Form zusammen wie oben für SI dargelegt wurde. Die reduzierte Genauigkeit der finiten Differenzen macht sich in den Ergebnissen klar bemerkbar. Sie bedingt für  $\text{TOL}_{\text{Nom}} \geq 10^{-4}$  relativ schnelle Abbrüche der SQP-Iterationen, die sich entsprechend auf die Zielfunktions- und Parameterwerte niederschlagen. Nur hohe Genauigkeitsvorgaben führen zu Werten in der Zielfunktion, die mit SI für Durchschnittsanforderungen erreicht werden. Die dafür nötigen Berechnungszeiten sind mit SI zum Teil erheblich kleiner. Ein Blick auf die errechneten optimalen Parameterwerte verdeutlicht, dass Schwierigkeiten bestehen, die Parameter des Robotermodells mit Hilfe von finiten Differenzen zu identifizieren.

Zusammenfassend lässt sich feststellen, dass wir einen Ausschnitt einer umfassenden Parameteridentifizierung an einem industriell eingesetzten Robotermodell mit Unstetigkeiten nachvollziehen können. Unter Einbindung der entwickelten Softwareumgebung FISEMO gelingt es für das eingesetzte SQP-Verfahren, genauere Gradienten zu erzeugen, als mit finiten Differenzen erreicht werden. Die Vorteile dieses Berechnungsansatzes können an dem komplexen Modell und seiner Parameteridentifizierung nachgewiesen werden. Diese Aussage gilt ganz besonders für höhere Genauigkeitsansprüche an das Optimierungsergebnis.

## 7 Zusammenfassung und Ausblick

Mathematische Modelle technischer Systeme bilden eine wichtige Säule in der Forschung und in der Entwicklung neuer Produkte. Die Systemmodellierung, -simulation und -optimierung dynamischer Prozesse werden auf immer komplexer vernetzte Problemstellungen angewendet. Daher spielen multidisziplinäre Untersuchungen, beispielsweise zur Reglerauslegung oder zur Bewertung von dynamischen Systemeigenschaften, eine entscheidende Rolle. Zusätzlich zur Modellierungsbreite steht die Frage nach der angemessenen Modellierungstiefe im Raum. Werden gewisse physikalische Gesetzmäßigkeiten vereinfacht oder ähnliche Einflüsse zusammengefasst, so entstehen in diesem Zusammenhang häufig Modellgleichungen, die Unstetigkeiten und Umschaltvorgänge enthalten.

Diese unstetigen mathematischen Systemmodelle können grundlegend andere theoretischen Eigenschaften besitzen. Auch für die numerische Lösung und Optimierung stellen strukturvariable Gleichungen eine Herausforderung dar, weil für viele Verfahren mehrfach differenzierbare Funktionen vorausgesetzt werden. Ein üblicher Ansatz zur numerischen Zeitintegration unstetiger Modellgleichungen verwendet Schaltfunktionen, die über ihre Nullstellen die exakten Ereigniszeitpunkte der Unstetigkeitsstellen definieren und damit das Integrationsintervall in Teilintervalle zerlegen. Innerhalb der Teilintervalle können dann hinreichend glatte Modellgleichungen mit Standardalgorithmen integriert werden, bevor jeweils auf das zeitlich anschließende Teilsystem umgeschaltet wird.

Die vorliegende Arbeit leistet sowohl zur Theorie unstetiger Systeme als auch zur numerischen Problemlösung neue Beiträge. Im theoretischen Teil wird eine Lösungstheorie, die Parametersensitivitäten mit einschließt, für unstetige Systemmodelle entwickelt. Besonderes Augenmerk liegt darauf, die verallgemeinerten Lösungen nach Filippov als festen Bestandteil in die Ausführungen aufzunehmen. Es gelingt, Lösungen für Systeme mit nichtglatten Schaltfunktionen zu konstruieren. Die Anregung, diese Klasse von unstetigen Systemen zu studieren, geht auf ein anwendungsbezogenes mechanisches Beispiel zurück. Praktische Anwendungen erfordern zusätzlich die Analyse von Unstetigkeiten, die durch mehrere zeitgleich aktiv werdende Schaltfunktionen hervorgerufen werden. Ihre Auswirkungen auf die Lösungsdefinition – auch im Hinblick auf Filippov-Lösungen – sind Bestandteil der aufgezeigten Lösungstheorie.

Die Sensitivitätsanalyse baut auf den Ergebnissen des konstruktiven Lösungsnachweises auf. Die Sensitivitätsgleichungen definieren die Sensitivitätszustände zwischen den Umschaltzeitpunkten. Während des Umschaltens können diese Zustände unstetige Änderungen erfahren, die dazu führen, dass die Sensitivitäten der zusammengesetzten Lösung am Ereignispunkt unter Umständen nicht existieren. Die Formeln der abschnittsweise festgelegten Parametersensitivitäten übertragen sich in natürlicher Weise auf die der Filippov-Lösung. Neue Ergebnisse werden zur Sensitivitätsanalyse der unstetigen Systeme mit mehreren zeitgleich aktiven Schaltfunktionen präsentiert. Unter gewissen Voraussetzungen an

die Schaltstruktur werden Wege aufgezeigt, die Existenz der Sensitivitäten nachzuweisen und diese zu berechnen. Gegenbeispiele belegen die Notwendigkeit der Voraussetzungen und demonstrieren Situationen, in denen die Sensitivitäten nach dem Umschalten nicht mehr existieren.

Der numerische Teil der Arbeit stellt die Softwareumgebung FISEMO vor, die einen Teil der theoretischen Erkenntnisse umsetzt. Grundlage dafür ist die Modellierungssprache Modelica – eines der Standardwerkzeuge zur systemdynamischen Analyse multiphysikalischer Systeme in Forschung und praktischen Anwendungen. Wesentliche Bausteine von FISEMO sind der von Dymola automatisch erzeugte C-Code zum Modelica-Modell, das Paket DASPK3.1 zur Integration der Sensitivitätsgleichungen sowie die neu implementierten Berechnungen an den Ereignispunkten. Diese Berechnungen dienen dazu, den Übergang zu Filippov-Lösungen festzustellen und für unstetige Änderungen der Sensitivitätszustände die Anfangswerte für den nachfolgenden Zeitabschnitt neu zu setzen. Numerische Ergebnisse ausgewählter multidisziplinärer Modelica-Modelle aus der Elektromechanik weisen die Funktionalität der Software FISEMO nach. Die Genauigkeit der berechneten Lösungen und der Sensitivitäten liegen im Wesentlichen im Bereich der vorgegebenen Fehlertoleranzen.

Mit dem Einfachschießverfahren werden vier Parameter eines Industrierobotermodells erfolgreich identifiziert. Die Auswahl der Parameter stützt sich im Vorfeld der Optimierung auf verschiedene Kriterien, die allesamt mit Hilfe der Sensitivitäten des Modells berechnet werden. Die Achsenantriebe innerhalb des Robotermodells verursachen Haftreibungsphasen, deren Filippov-Lösungen von FISEMO ohne zusätzliche Modellinformationen numerisch berechnet werden. Die SQP-Optimierungsergebnisse werden für unterschiedliche Genauigkeitsvorgaben an die Integration gegenübergestellt. Sie beleuchten neben den Werten der Zielfunktion besonders die optimalen Parameter. Streuungen der identifizierten Parameterwerte, die von Unsicherheiten in den Messwerten der vergleichenden Größen verursacht werden, überlagern sich mit den numerischen Abweichungen auf Grund der Fehlertoleranzen. Die Ergebnisse illustrieren den erfolgreichen Einsatz der Methodik und der Software in einer komplexen Anwendung. Der abschließende Vergleich, die Optimierung mit finiten Differenzen zur Ableitungsberechnung durchzuführen, unterstreicht zusätzlich die Vorteile der in FISEMO implementierten Technik.

Weiterführende Arbeiten sind in verschiedene Richtungen denkbar. In den implementierten Berechnungsverfahren wären Effizienzsteigerungen möglich, wenn Strukturinformationen des Modells für die Auswertung von partiellen Ableitungen oder der rechten Seite für Filippov-Lösungen ausgenutzt würden. Dafür wäre es notwendig, die Strukturdaten im C-Code eines Modelica-Modells verfügbar zu machen. Die Sensitivitätsberechnungen könnten in der Optimalsteuerung von unstetigen multidisziplinären Modellen zu verbessertem Konvergenzverhalten der gradientenbasierten Optimierungsverfahren beitragen. Sinnvolle Erweiterungen des Modelica-Standards für eine anwenderfreundliche optimale Versuchsplanung zur Parameteridentifizierung könnten diesem Thema noch weiteres Gewicht geben. Darüber hinaus wäre es vorstellbar, die Modelica-Sprache allgemein mehr in die Richtungen Optimierung und Optimalsteuerung von unstetigen Systemmodellen weiterzuentwickeln. Die Sensitivitätsanalyse wäre hierfür ein geeignetes Bindeglied.

# A Programmablaufpläne

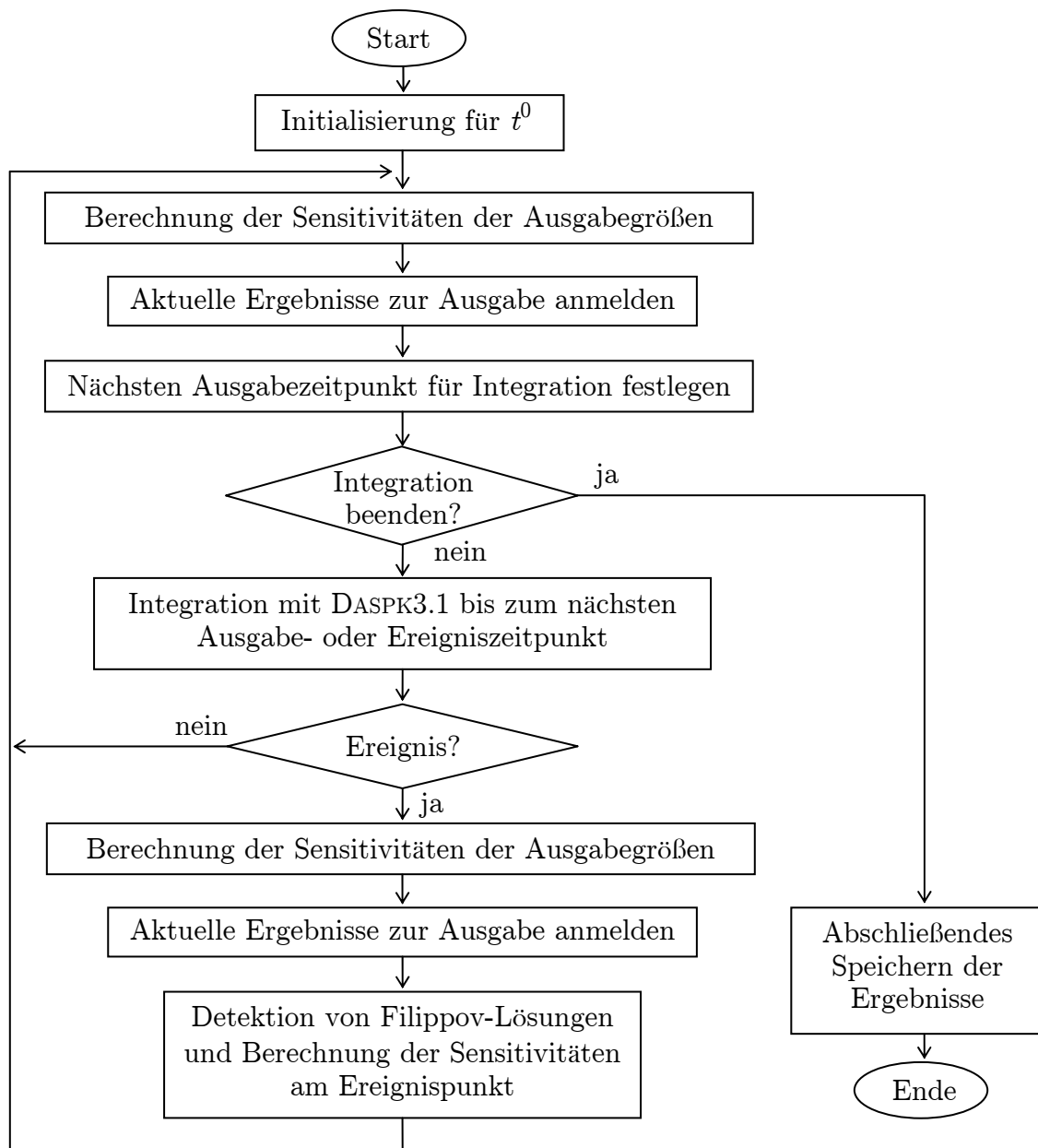


Abbildung A.1: Ablaufplan der Integratorsteuerung in der Umgebung FISEMO.

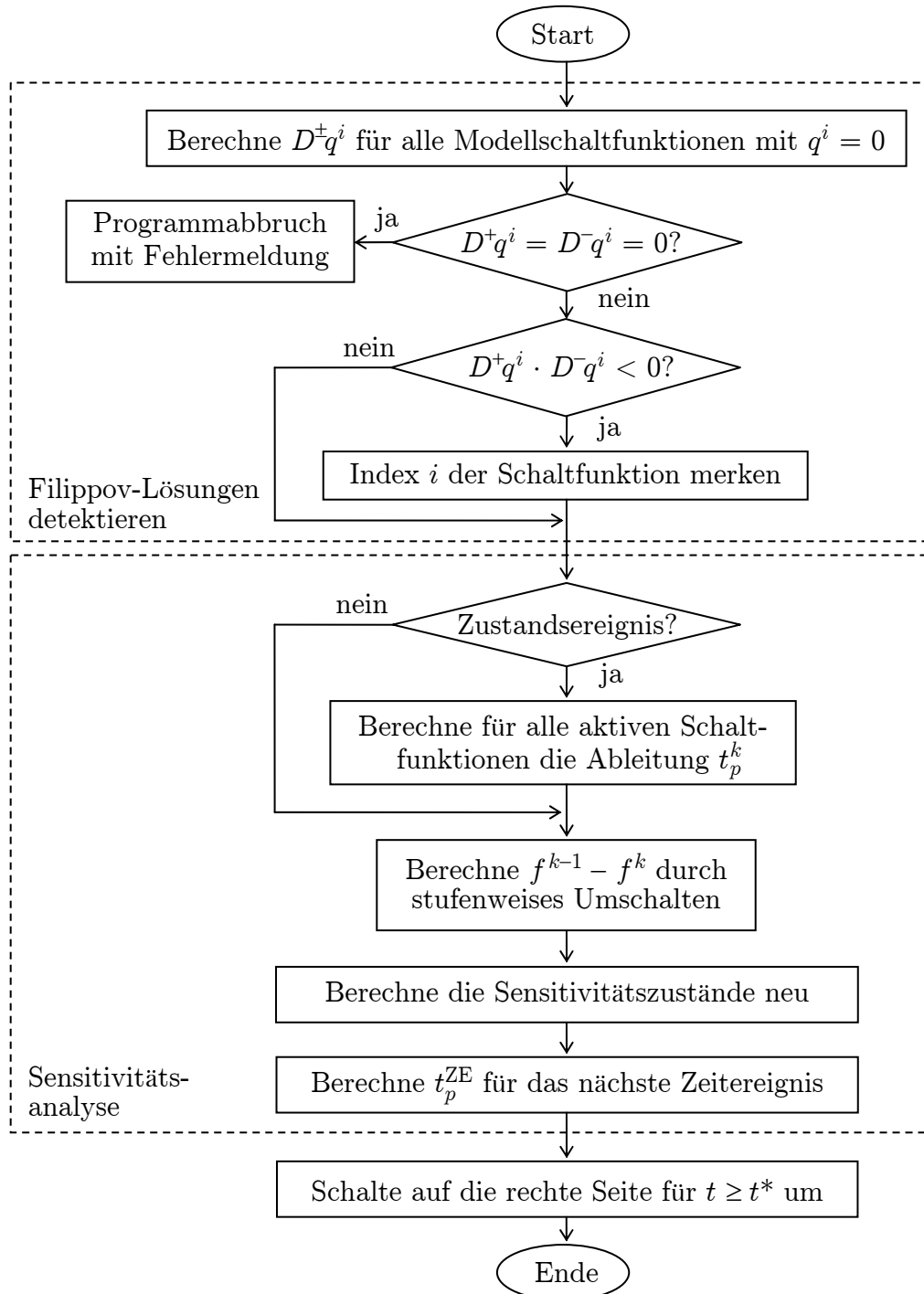


Abbildung A.2: Ablaufplan für die Berechnungen am Ereignispunkt. Aufgeteilt in die Erkennung von Filippov-Lösungen und die Sensitivitätsberechnungen.



## B Ergebnisse der Sensitivitätsanalysen

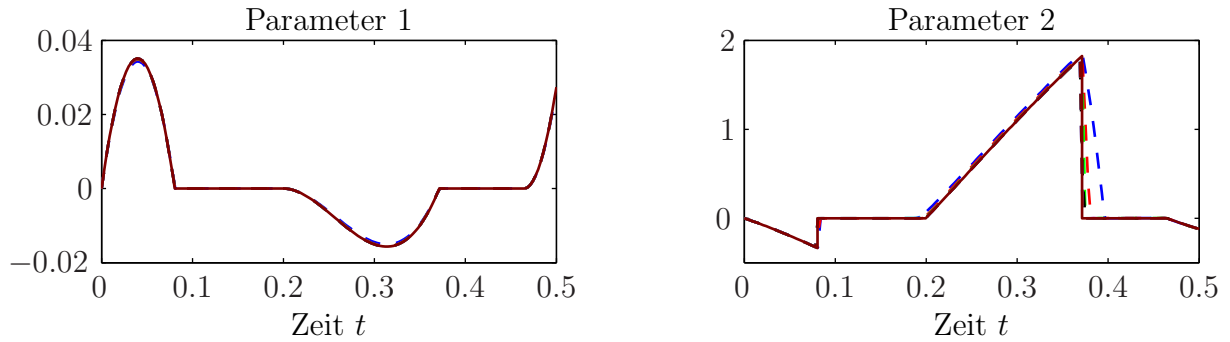


Abbildung B.1: Zeitverläufe der Sensitivitäten  $(\omega_1)_p$  von Beispiel 5.5. Externe Differenzenquotienten mit verschiedenen Vorgaben für  $\text{TOL}_{\text{FD}}$  im Vergleich zur Referenzlösung (braune Linie). Farbgebung nach Tabelle 5.2.

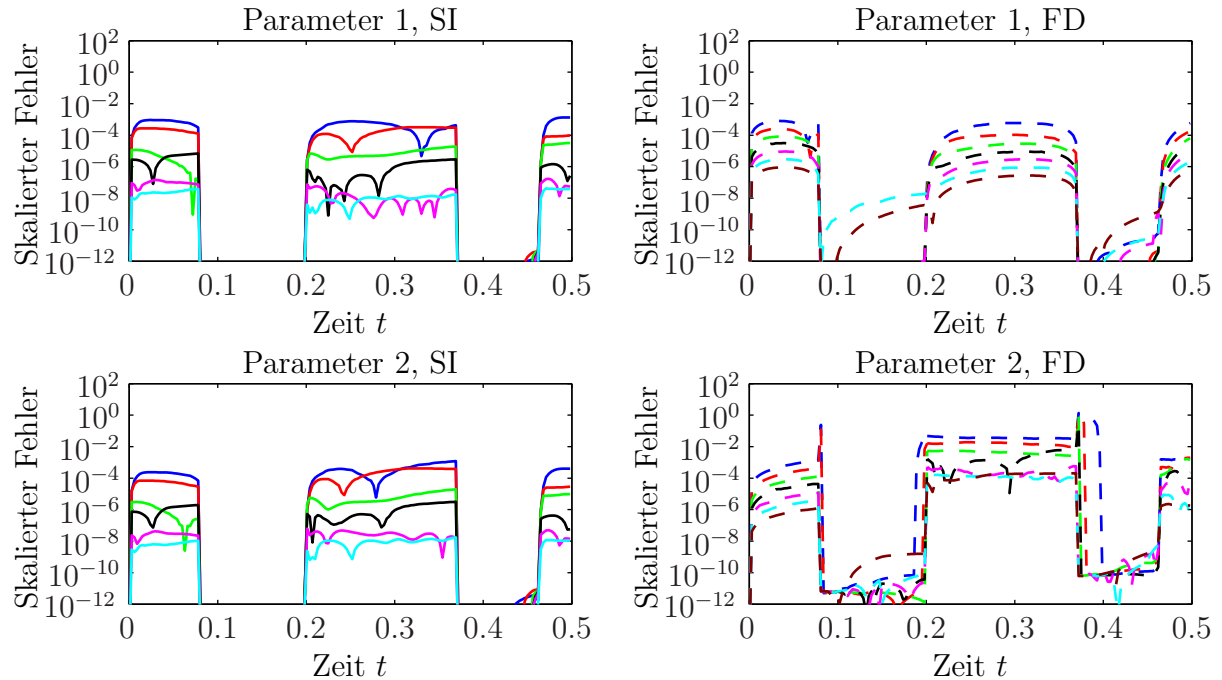


Abbildung B.2: Fehler in den Sensitivitäten  $(\omega_1)_p$  von Beispiel 5.5. Integration der Sensitivitäten (links) im Vergleich zu externen Differenzenquotienten (rechts). Farbgebung nach Tabelle 5.2.

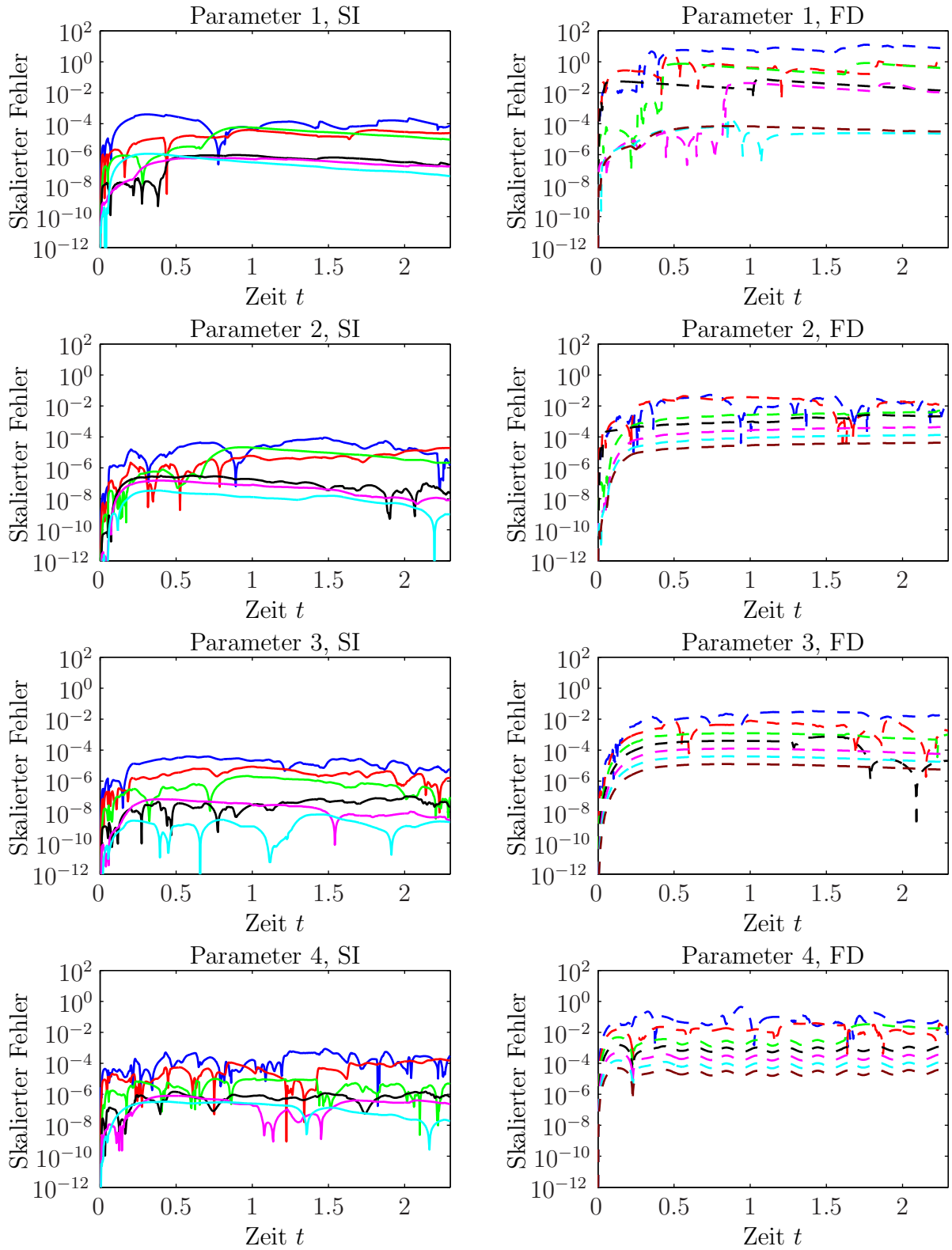


Abbildung B.3: Fehler in den Sensitivitäten  $\omega_p$  von Beispiel 5.4. Integration der Sensitivitäten (links) im Vergleich zu externen Differenzenquotienten (rechts). Verschiedene Vorgaben für  $\text{TOL}_{\text{SI}}$ ,  $\text{TOL}_{\text{FD}}$ . Farbgebung nach Tabelle 5.2.

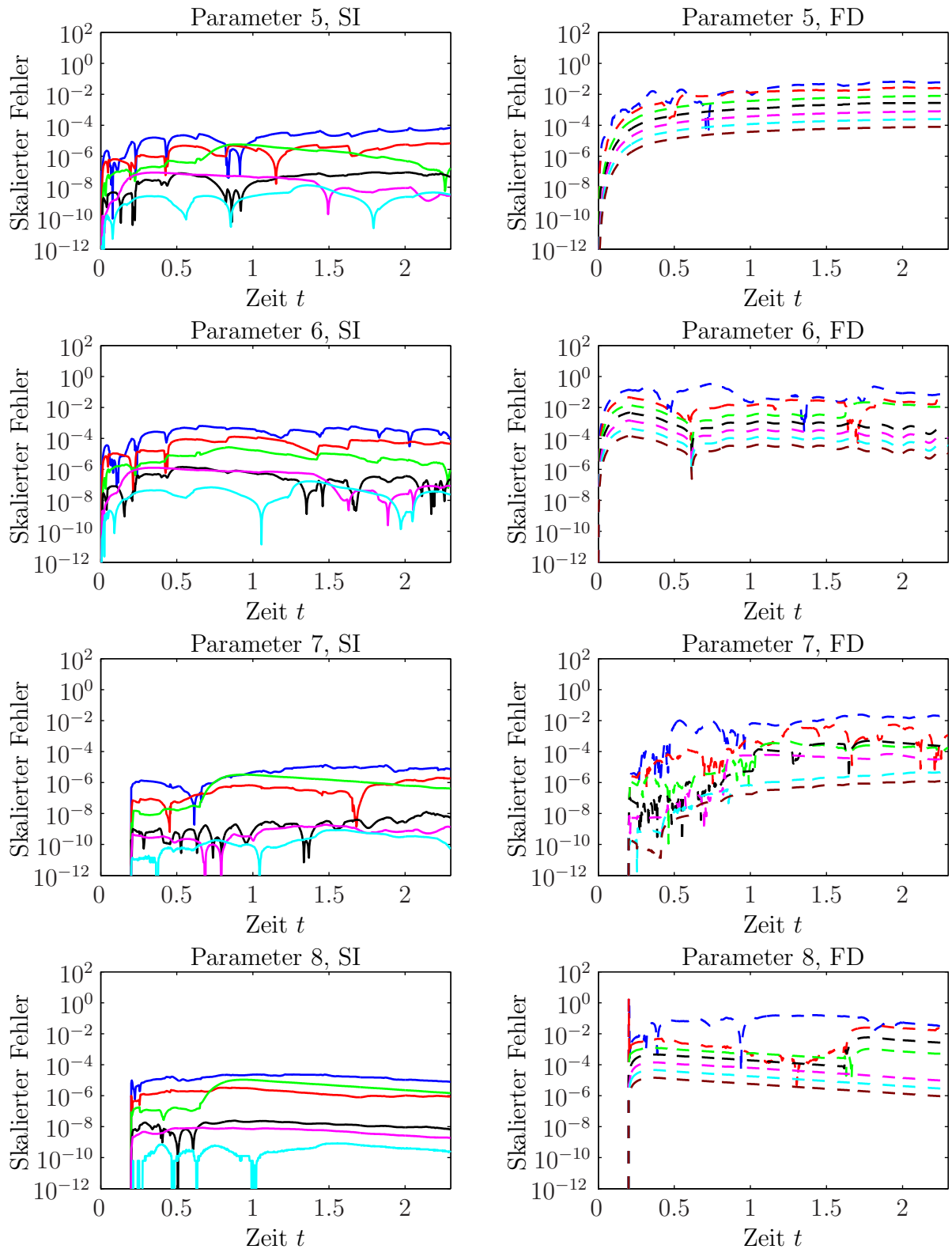


Abbildung B.4: Fehler in den Sensitivitäten  $\omega_p$  von Beispiel 5.4. Integration der Sensitivitäten (links) im Vergleich zu externen Differenzenquotienten (rechts). Verschiedene Vorgaben für  $\text{TOL}_{\text{SI}}$ ,  $\text{TOL}_{\text{FD}}$ . Farbgebung nach Tabelle 5.2.

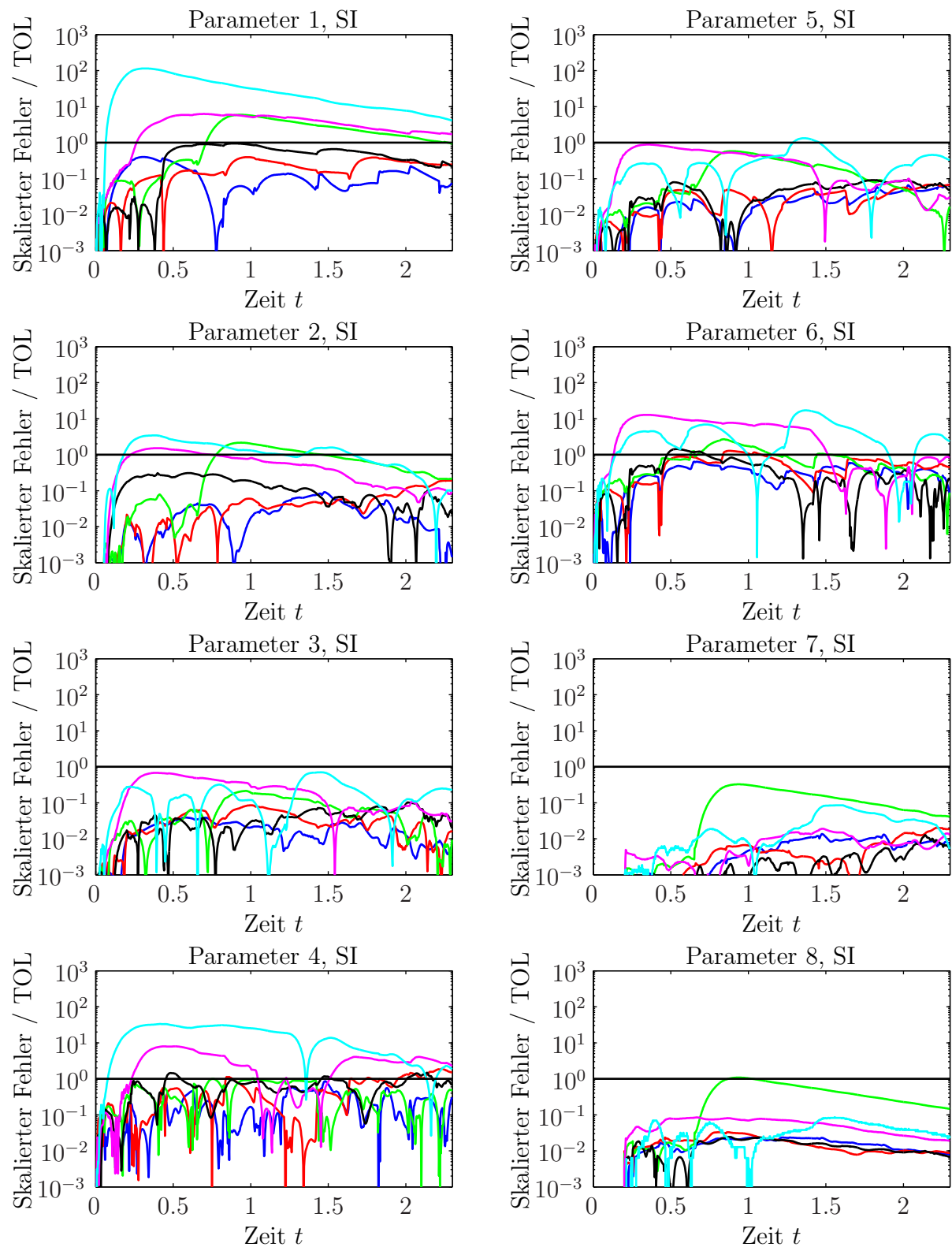


Abbildung B.5: Fehler in den Sensitivitäten  $\omega_p$  von Beispiel 5.4 bezogen auf die vorgegebene Fehlertoleranz  $TOL_{SI}$ . Integration der Sensitivitätsgleichungen mit unterschiedlichen Fehlertoleranzen  $TOL_{SI}$ . Farbgebung nach Tabelle 5.2.

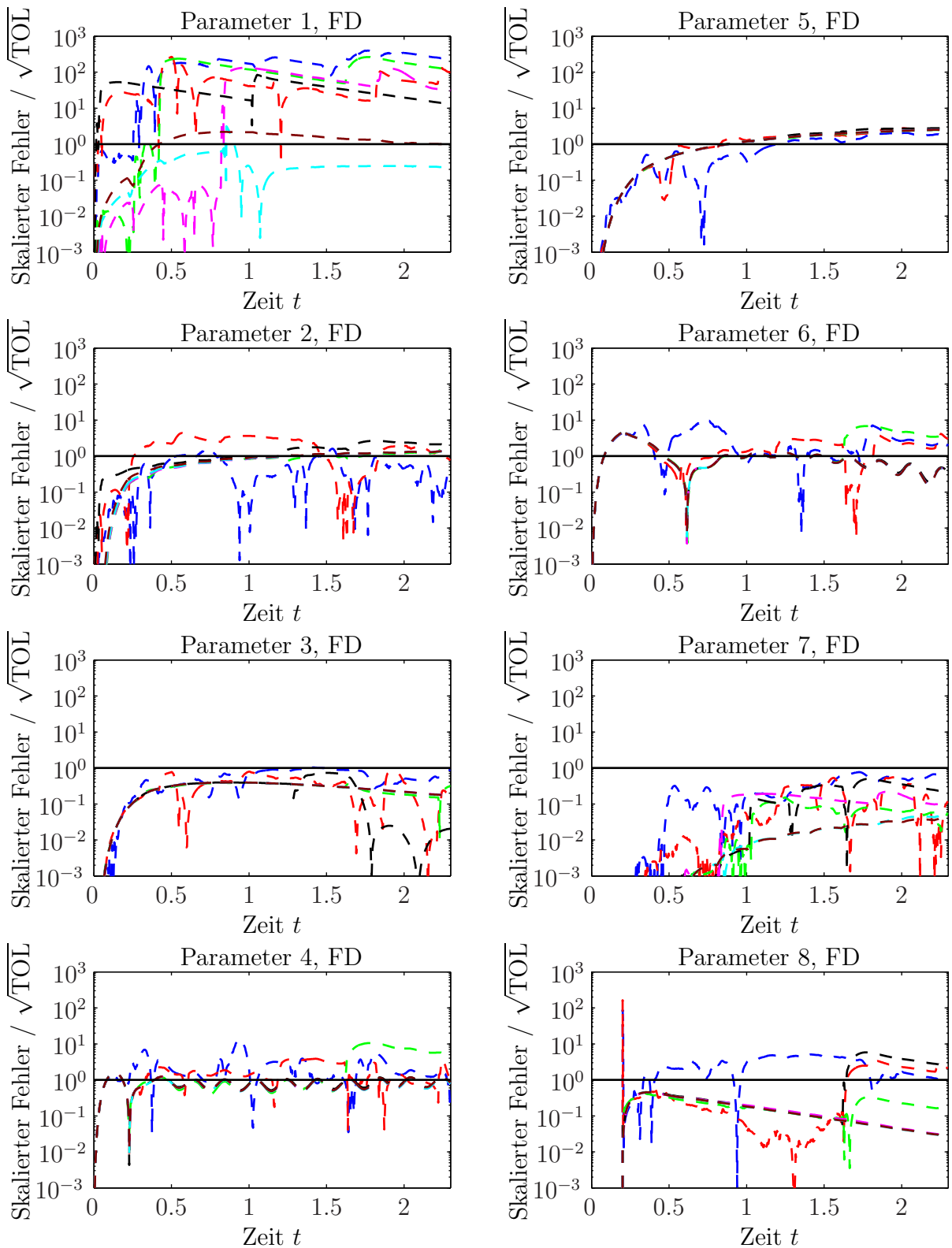


Abbildung B.6: Fehler in den Sensitivitäten  $\omega_p$  von Beispiel 5.4 bezogen auf die zu erwartende Genauigkeit  $\sqrt{\text{TOL}_{\text{FD}}}$ . Externe Differenzenquotienten mit unterschiedlichen Fehlertoleranzen  $\text{TOL}_{\text{FD}}$ . Farbgebung nach Tabelle 5.2.

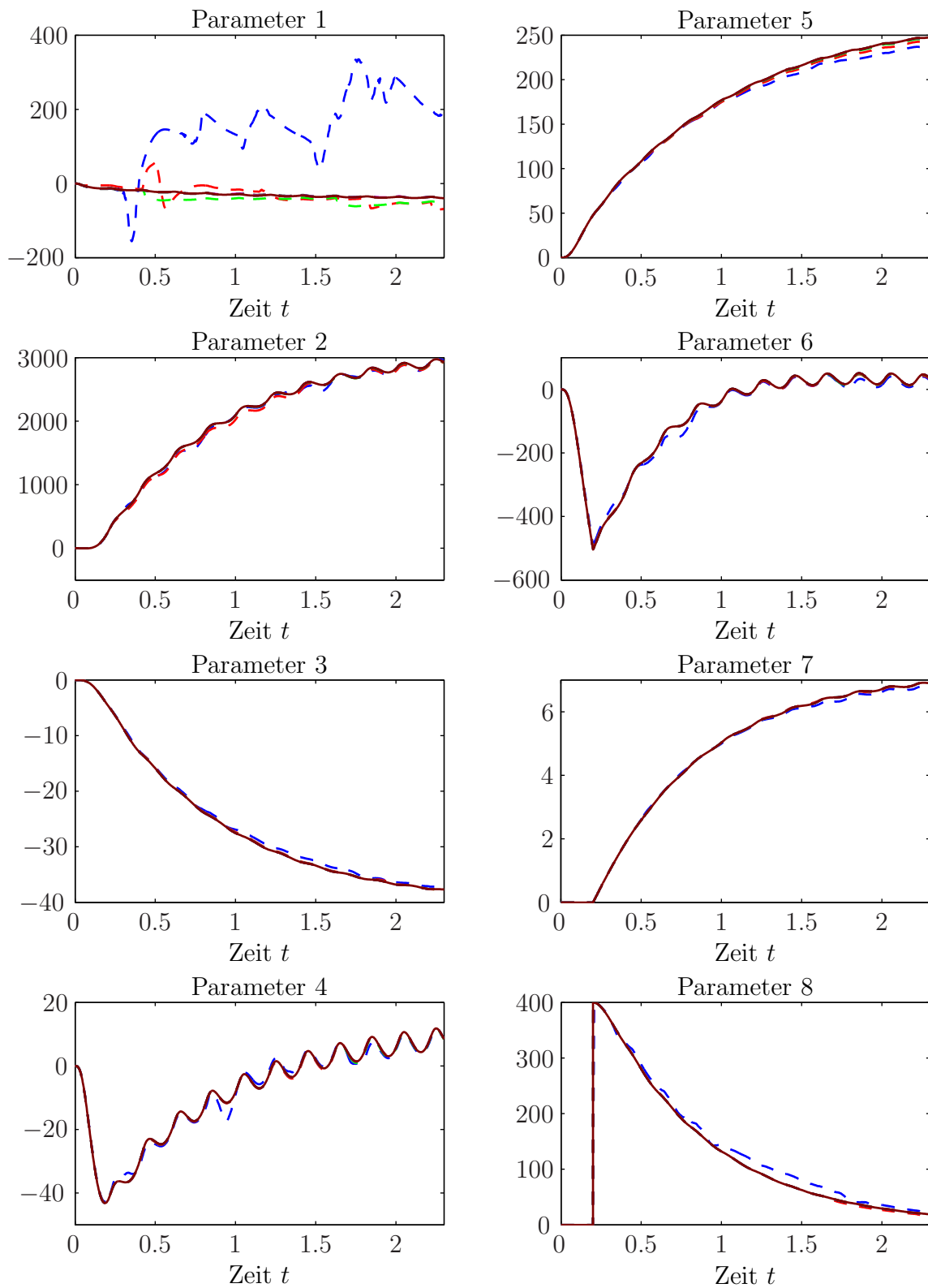


Abbildung B.7: Zeitverläufe der Sensitivitäten  $\omega_p$  von Beispiel 5.4. Externe Differenzenquotienten mit verschiedenen Vorgaben für  $TOL_{FD}$  im Vergleich zur Referenzlösung (braune Linie). Farbgebung nach Tabelle 5.2.

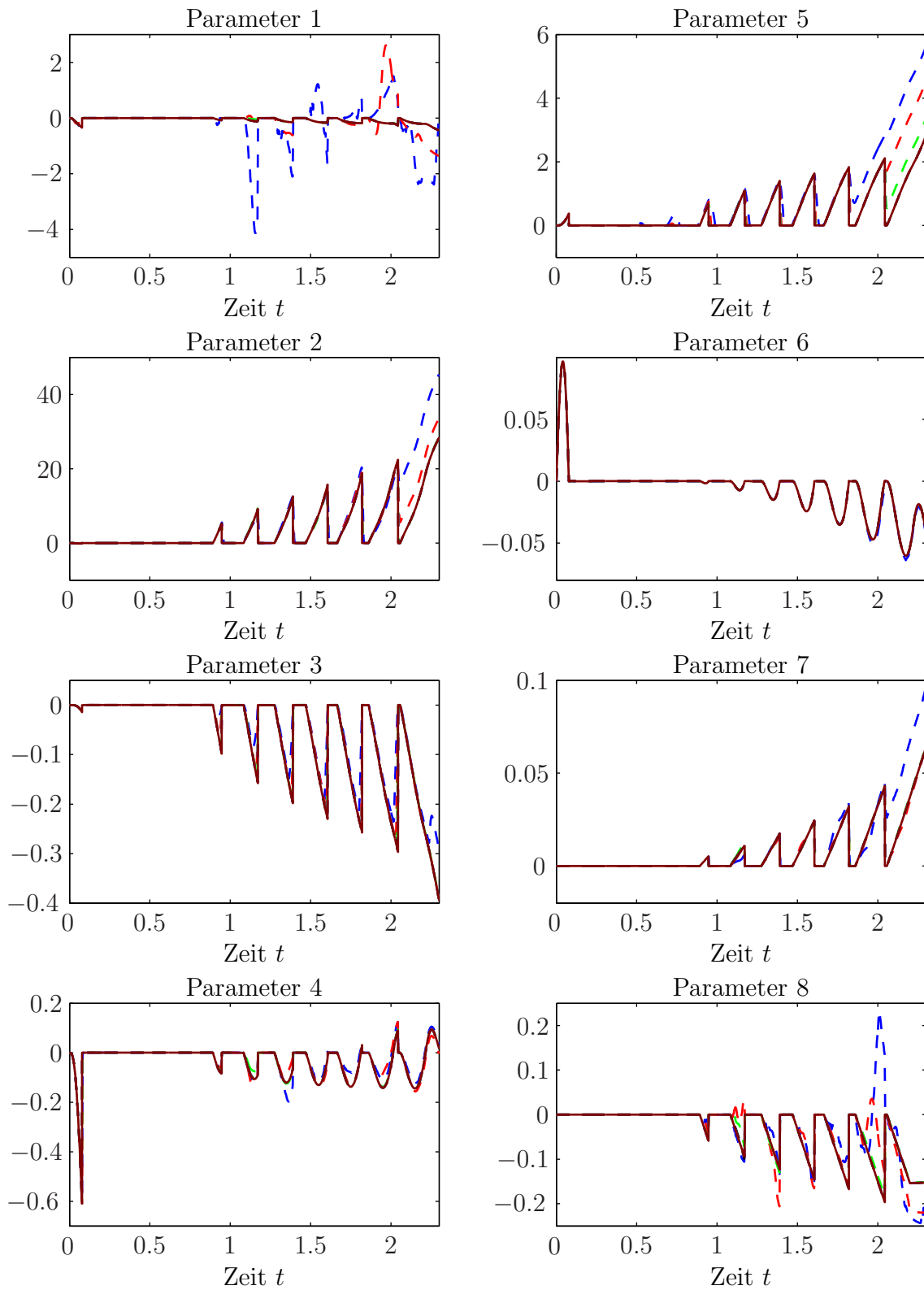


Abbildung B.8: Zeitverläufe der Sensitivitäten  $(\omega_2)_p$  von Beispiel 5.6. Externe Differenzenquotienten mit verschiedenen Vorgaben für  $TOL_{FD}$  im Vergleich zur Referenzlösung (braune Linie). Farbgebung nach Tabelle 5.2.

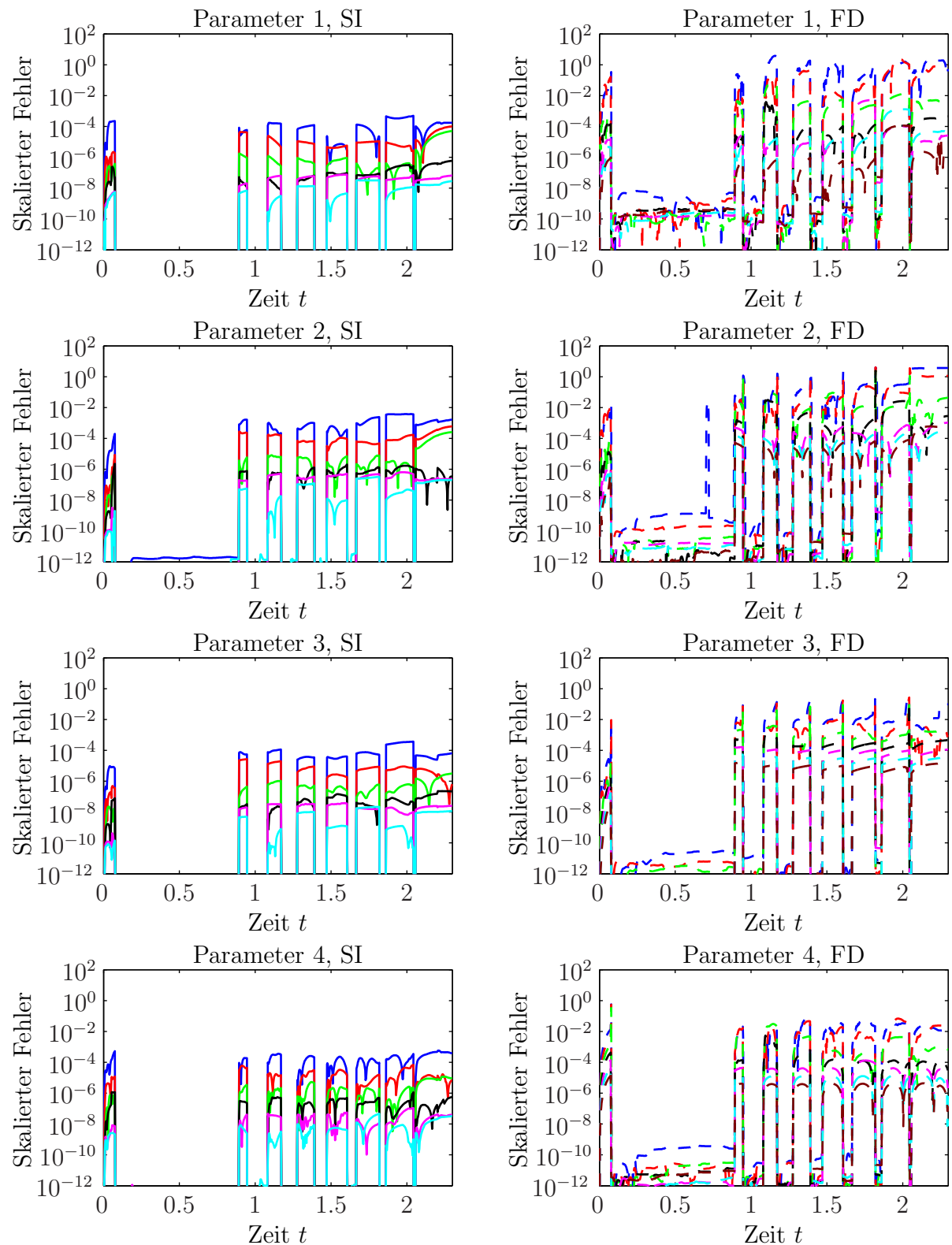


Abbildung B.9: Fehler in den Sensitivitäten  $(\omega_2)_p$  von Beispiel 5.6. Integration der Sensitivitäten (links) im Vergleich zu externen Differenzenquotienten (rechts). Verschiedene Vorgaben für TOL<sub>SI</sub>, TOL<sub>FD</sub>. Farbgebung nach Tabelle 5.2.



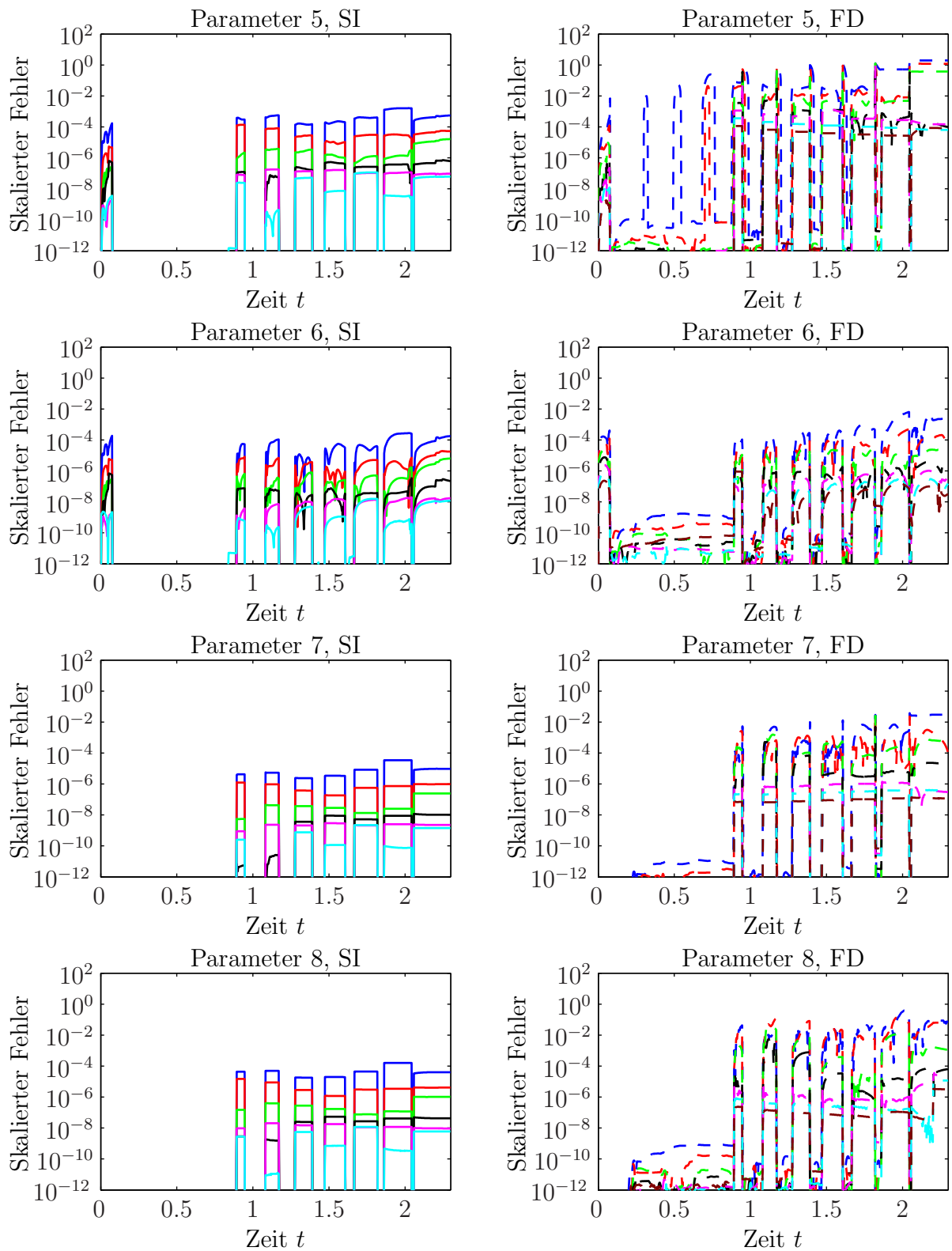


Abbildung B.10: Fehler in den Sensitivitäten  $(\omega_2)_p$  von Beispiel 5.6. Integration der Sensitivitäten (links) im Vergleich zu externen Differenzenquotienten (rechts). Farbgebung nach Tabelle 5.2.

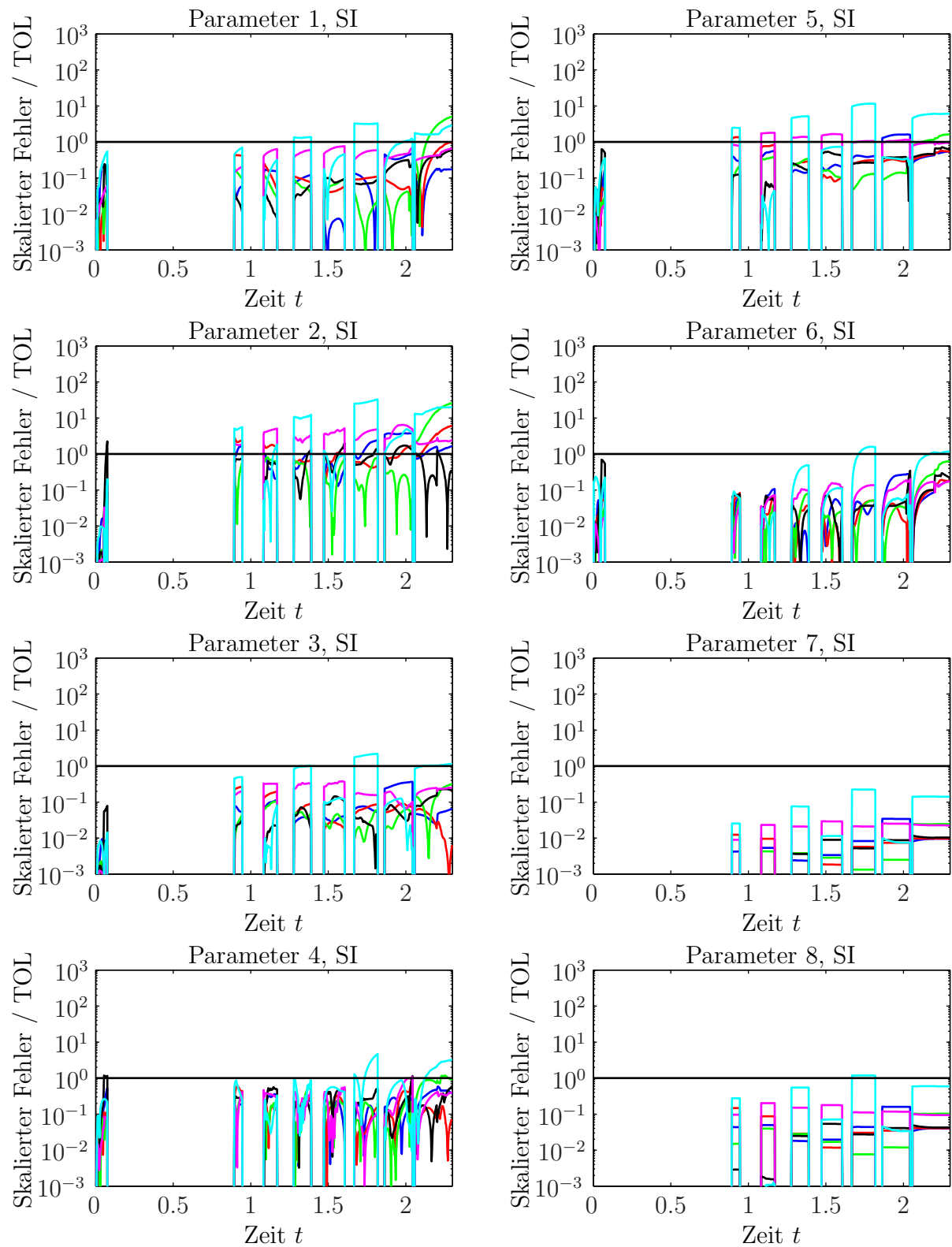


Abbildung B.11: Fehler in den Sensitivitäten  $(\omega_2)_p$  von Beispiel 5.6 bezogen auf die Fehlertoleranz  $\text{TOL}_{\text{SI}}$ . Integration der Sensitivitätsgleichungen mit unterschiedlichen Fehlertoleranzen  $\text{TOL}_{\text{SI}}$ . Farbgebung nach Tabelle 5.2.

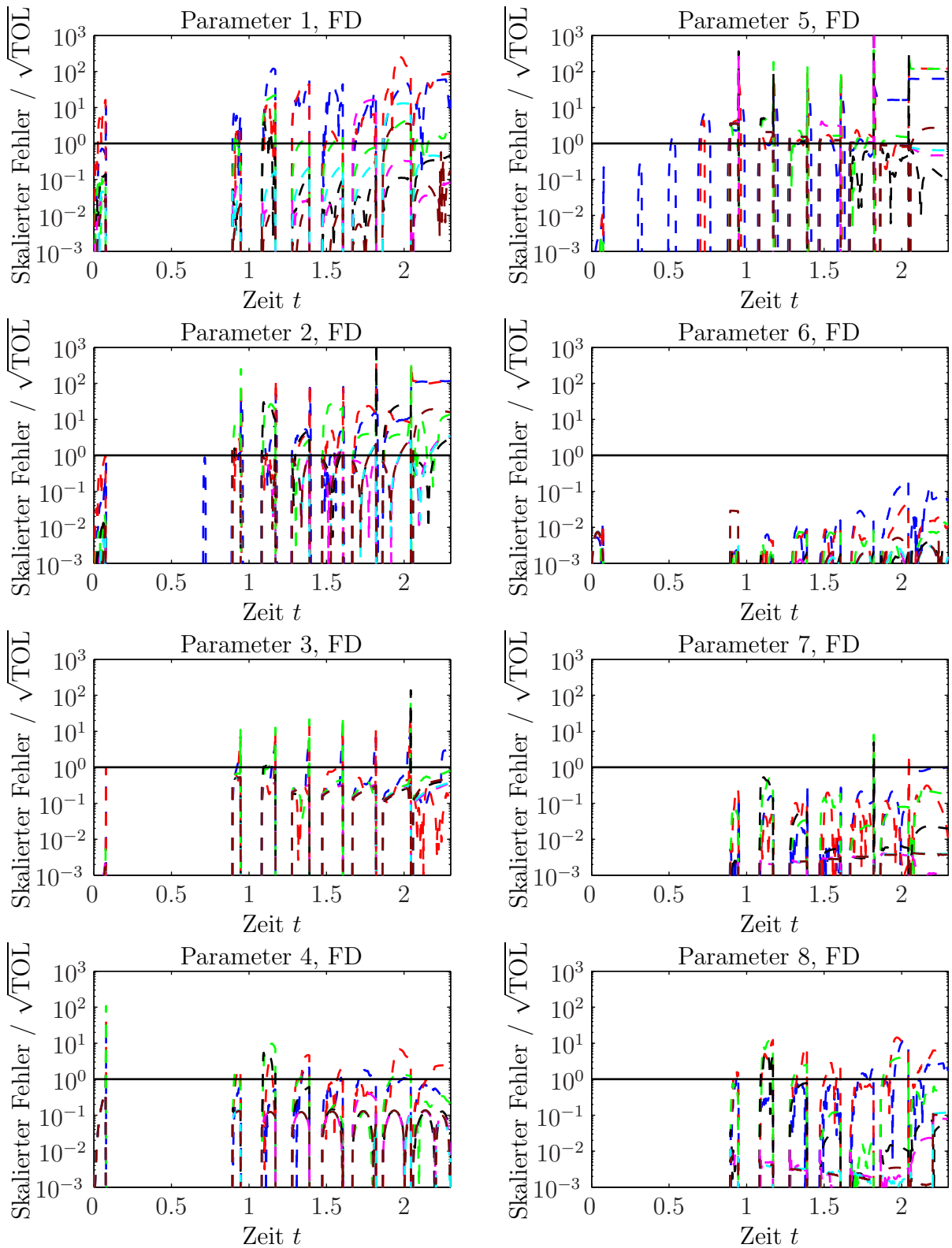


Abbildung B.12: Fehler in den Sensitivitäten  $(\omega_2)_p$  von Beispiel 5.6 bezogen auf die zu erwartende Genauigkeit  $\sqrt{\text{TOL}_{\text{FD}}}$ . Externe Differenzenquotienten mit unterschiedlichen Fehlertoleranzen  $\text{TOL}_{\text{FD}}$ . Farbgebung nach Tabelle 5.2.

## C Ergebnisse der Parameteridentifizierung

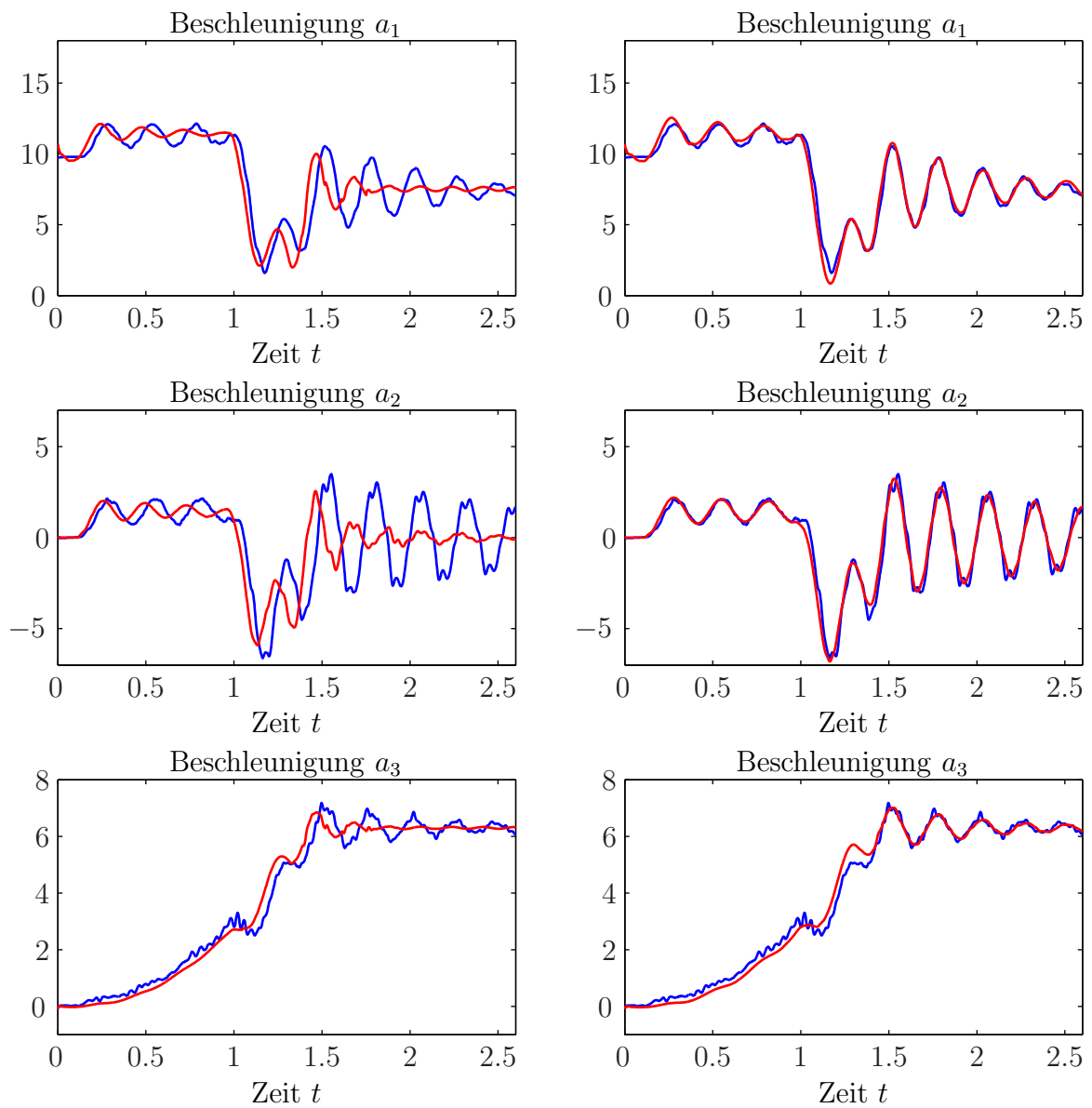


Abbildung C.1: Zeitverläufe der Beschleunigungen für  $p = p^0$  (links) und  $p = p^{\text{Ref}}$  (rechts). Fehlertoleranz  $\text{TOL} = 10^{-7}$ . **Blau:** Messungen. **Rot:** Simulationsergebnisse.

Tabelle C.1: Ergebnisse der Optimierungsläufe mit unterschiedlichen Vorgaben der Fehler-toleranzen  $\text{TOL}_{\text{SI}}$  und  $\text{TOL}_{\text{Nom}}$ . Integration der Sensitivitäten mit FISEMO.

$\text{TOL}_{\text{SI}}$	$\text{TOL}_{\text{Nom}}$	$\#C$	$\#C_p$	$C(p^*)$	$i = 1$	$\bar{p}_i^*$			Rechenzeit in min
$10^{-2}$	$10^{-2}$	61	13	2.41	0.33	0.16	-0.12	0.03	32
$10^{-2}$	$10^{-3}$	75	17	1.27	0.32	0.10	-0.08	-0.08	47
$10^{-2}$	$10^{-4}$	83	30	1.02	0.31	0.51	-0.11	0.00	84
$10^{-2}$	$10^{-5}$	71	25	1.03	0.32	0.18	-0.12	0.00	91
$10^{-2}$	$10^{-6}$	66	26	1.03	0.32	0.19	-0.11	-0.02	96
$10^{-2}$	$10^{-7}$	67	26	1.03	0.32	0.19	-0.11	-0.02	197
$10^{-3}$	$10^{-2}$	35	6	2.82	0.50	-1.00	-0.39	1.00	19
$10^{-3}$	$10^{-3}$	70	12	1.23	0.34	-0.12	0.01	-0.18	45
$10^{-3}$	$10^{-4}$	55	16	1.02	0.32	0.20	-0.12	0.01	58
$10^{-3}$	$10^{-5}$	47	21	1.00	0.34	-0.07	-0.12	0.01	68
$10^{-3}$	$10^{-6}$	47	21	1.00	0.34	-0.07	-0.12	0.01	75
$10^{-3}$	$10^{-7}$	50	21	1.00	0.34	-0.07	-0.12	0.01	154
$10^{-4}$	$10^{-2}$	36	5	2.47	0.42	-0.49	-0.12	0.09	24
$10^{-4}$	$10^{-3}$	83	26	1.26	0.33	0.16	-0.13	0.03	81
$10^{-4}$	$10^{-4}$	47	23	1.02	0.33	0.20	-0.11	0.00	76
$10^{-4}$	$10^{-5}$	46	22	1.01	0.32	0.20	-0.11	0.00	88
$10^{-4}$	$10^{-6}$	67	36	0.99	0.37	-0.29	-0.11	0.00	143
$10^{-4}$	$10^{-7}$	76	36	0.99	0.37	-0.29	-0.11	0.00	273
$10^{-5}$	$10^{-2}$	43	7	2.41	0.30	0.48	-0.26	0.38	36
$10^{-5}$	$10^{-3}$	16	4	5.32	1.00	-0.06	-0.18	0.11	21
$10^{-5}$	$10^{-4}$	47	22	1.02	0.38	-0.41	-0.11	-0.01	109
$10^{-5}$	$10^{-5}$	47	21	0.99	0.38	-0.41	-0.11	-0.01	122
$10^{-5}$	$10^{-6}$	47	22	0.99	0.38	-0.41	-0.11	-0.01	133
$10^{-5}$	$10^{-7}$	46	21	0.99	0.38	-0.41	-0.11	-0.01	211
$10^{-6}$	$10^{-2}$	48	8	2.40	0.36	-0.32	-0.28	0.44	50
$10^{-6}$	$10^{-3}$	44	11	1.71	0.51	-1.00	0.02	-0.16	74
$10^{-6}$	$10^{-4}$	51	25	1.02	0.38	-0.42	-0.11	-0.01	153
$10^{-6}$	$10^{-5}$	42	25	0.99	0.38	-0.42	-0.11	-0.01	160
$10^{-6}$	$10^{-6}$	41	22	0.99	0.38	-0.42	-0.11	-0.01	151
$10^{-6}$	$10^{-7}$	41	22	0.99	0.38	-0.42	-0.11	-0.01	227
$10^{-7}$	$10^{-2}$	69	20	2.30	0.36	0.25	-0.08	-0.05	188
$10^{-7}$	$10^{-3}$	81	23	1.30	0.39	-0.43	-1.00	-0.02	219
$10^{-7}$	$10^{-4}$	42	22	1.02	0.38	-0.42	-0.11	-0.01	207
$10^{-7}$	$10^{-5}$	44	25	0.99	0.38	-0.41	-0.11	-0.01	249
$10^{-7}$	$10^{-6}$	46	25	0.99	0.38	-0.41	-0.11	-0.01	256
$10^{-7}$	$10^{-7}$	58	26	0.99	0.38	-0.42	-0.11	-0.01	380

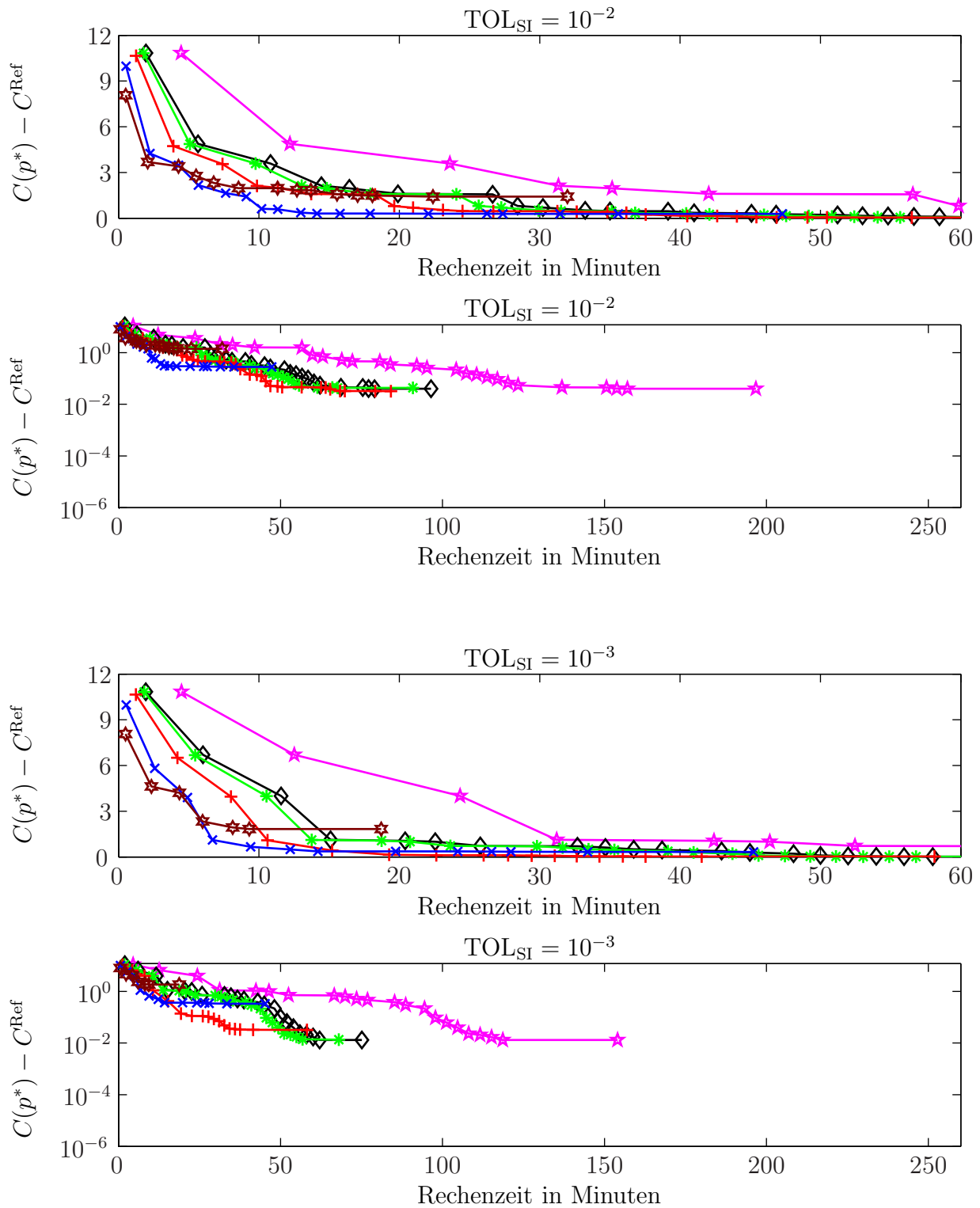


Abbildung C.2: Differenz zwischen Zielfunktion und Referenzwert in Abhängigkeit der benötigten Rechenzeit während der SQP-Iterationen für TOL<sub>SI</sub> = 10<sup>-2</sup>, 10<sup>-3</sup>. Lineare Skalierung des Fehlers für die ersten 60 min. Logarithmische Skalierung in der globalen Sicht. Farbgebung für TOL<sub>Nom</sub> nach Abbildung 6.2.

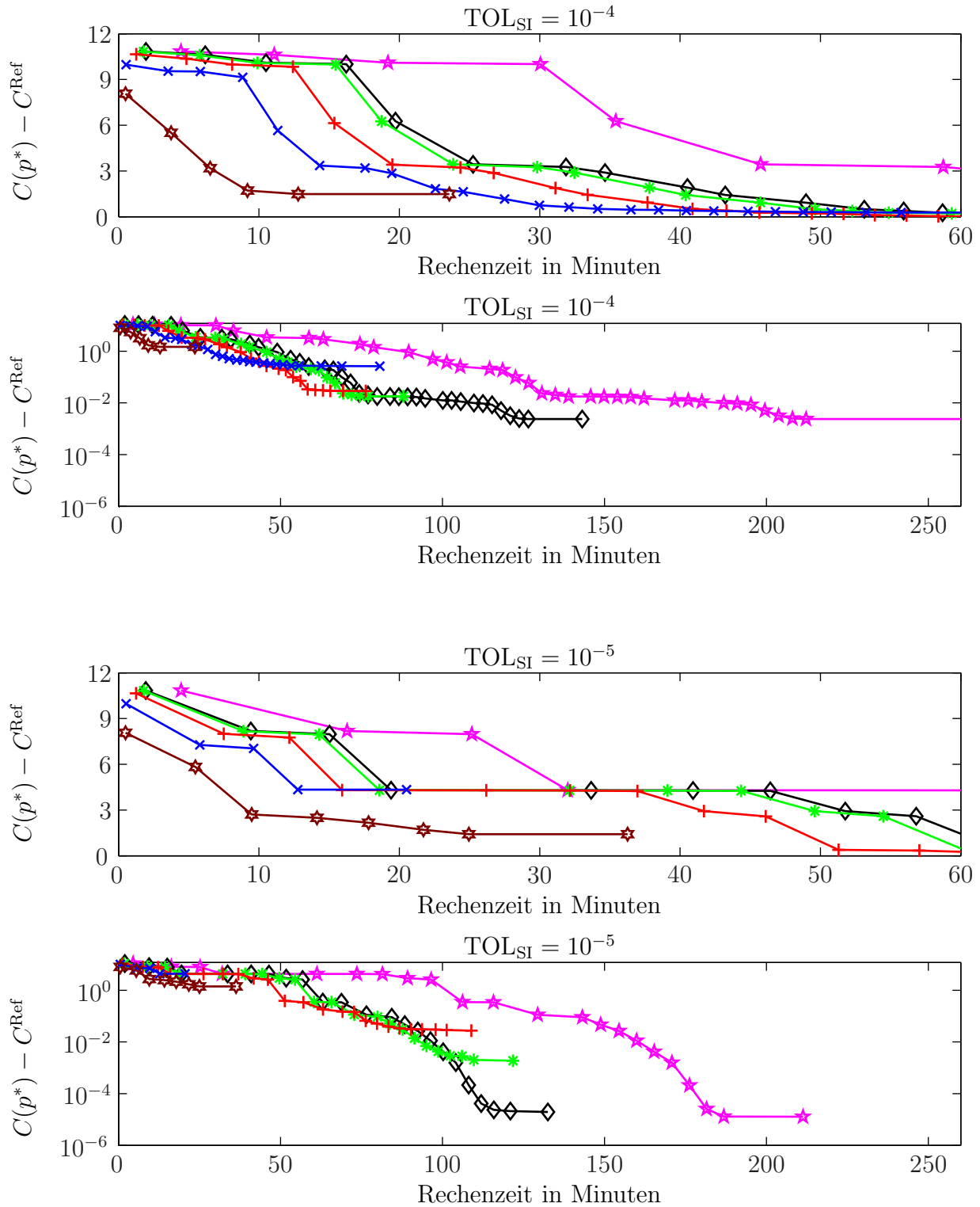


Abbildung C.3: Differenz zwischen Zielfunktion und Referenzwert in Abhängigkeit der benötigten Rechenzeit während der SQP-Iterationen für  $\text{TOL}_{\text{SI}} = 10^{-4}$ ,  $10^{-5}$ . Lineare Skalierung des Fehlers für die ersten 60 min. Logarithmische Skalierung in der globalen Sicht. Farbgebung für  $\text{TOL}_{\text{Nom}}$  nach Abbildung 6.2.

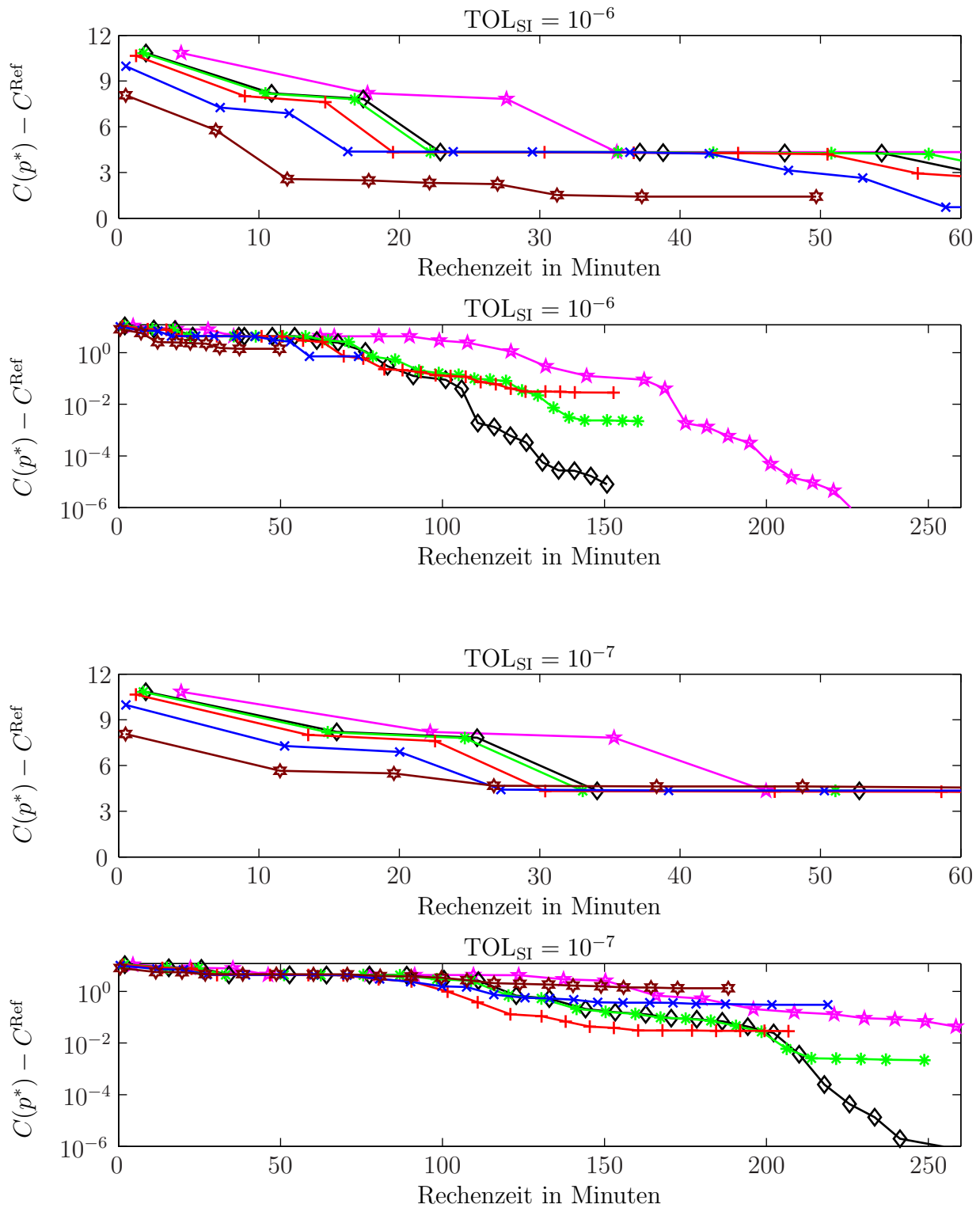


Abbildung C.4: Differenz zwischen Zielfunktion und Referenzwert in Abhängigkeit der benötigten Rechenzeit während der SQP-Iterationen für TOL<sub>SI</sub> = 10<sup>-6</sup>, 10<sup>-7</sup>. Lineare Skalierung des Fehlers für die ersten 60 min. Logarithmische Skalierung in der globalen Sicht. Farbgebung für TOL<sub>Nom</sub> nach Abbildung 6.2.



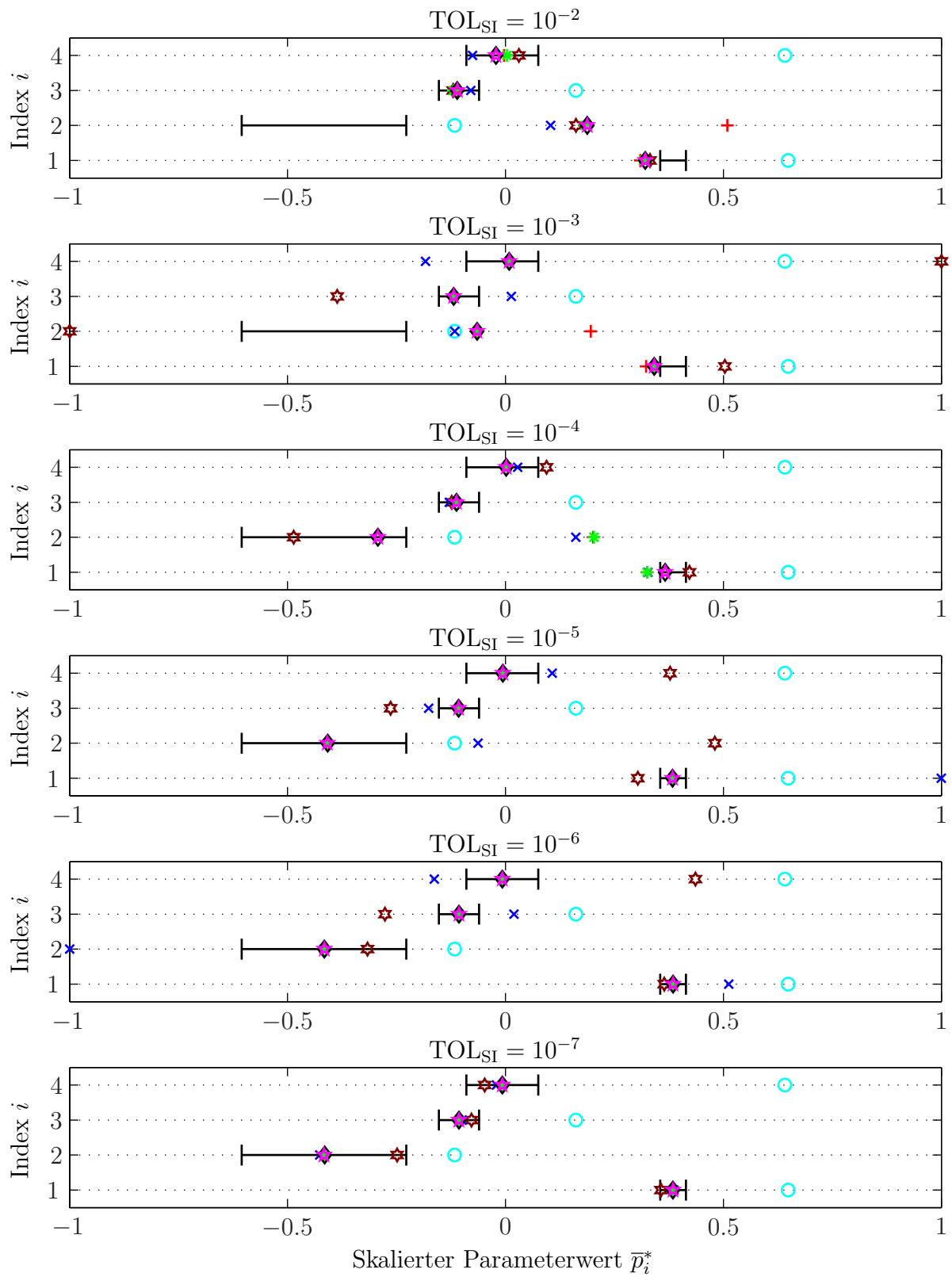


Abbildung C.5: Identifizierte Parameterwerte in Abhängigkeit von  $TOL_{SI}$  und  $TOL_{Nom}$ . Startwerte  $\bar{p}^0$  sind mit  $\circ$  gekennzeichnet. Farbgebung für  $TOL_{Nom}$  nach Abbildung 6.2.

Tabelle C.2: Ergebnisse der Optimierungsläufe mit unterschiedlichen Vorgaben der Fehler-toleranzen  $\text{TOL}_{\text{Nom}}$ . Finite Differenzenapproximation der Sensitivitäten.

$\text{TOL}_{\text{Nom}}$	$\#C$	$\#C_p$	$C(p^*)$	$i = 1$	2	3	4	Rechenzeit in min
$10^{-2}$	44	4	6.18	-0.03	-0.23	-0.26	0.51	22
$10^{-3}$	25	2	8.48	0.24	-0.27	0.05	0.68	16
$10^{-4}$	44	8	1.55	0.44	-0.69	-0.30	0.42	66
$10^{-5}$	60	12	1.17	0.37	-0.42	-0.22	0.18	124
$10^{-6}$	41	14	1.01	0.34	-0.07	-0.11	-0.01	131
$10^{-7}$	58	29	1.01	0.31	0.43	-0.09	-0.03	543

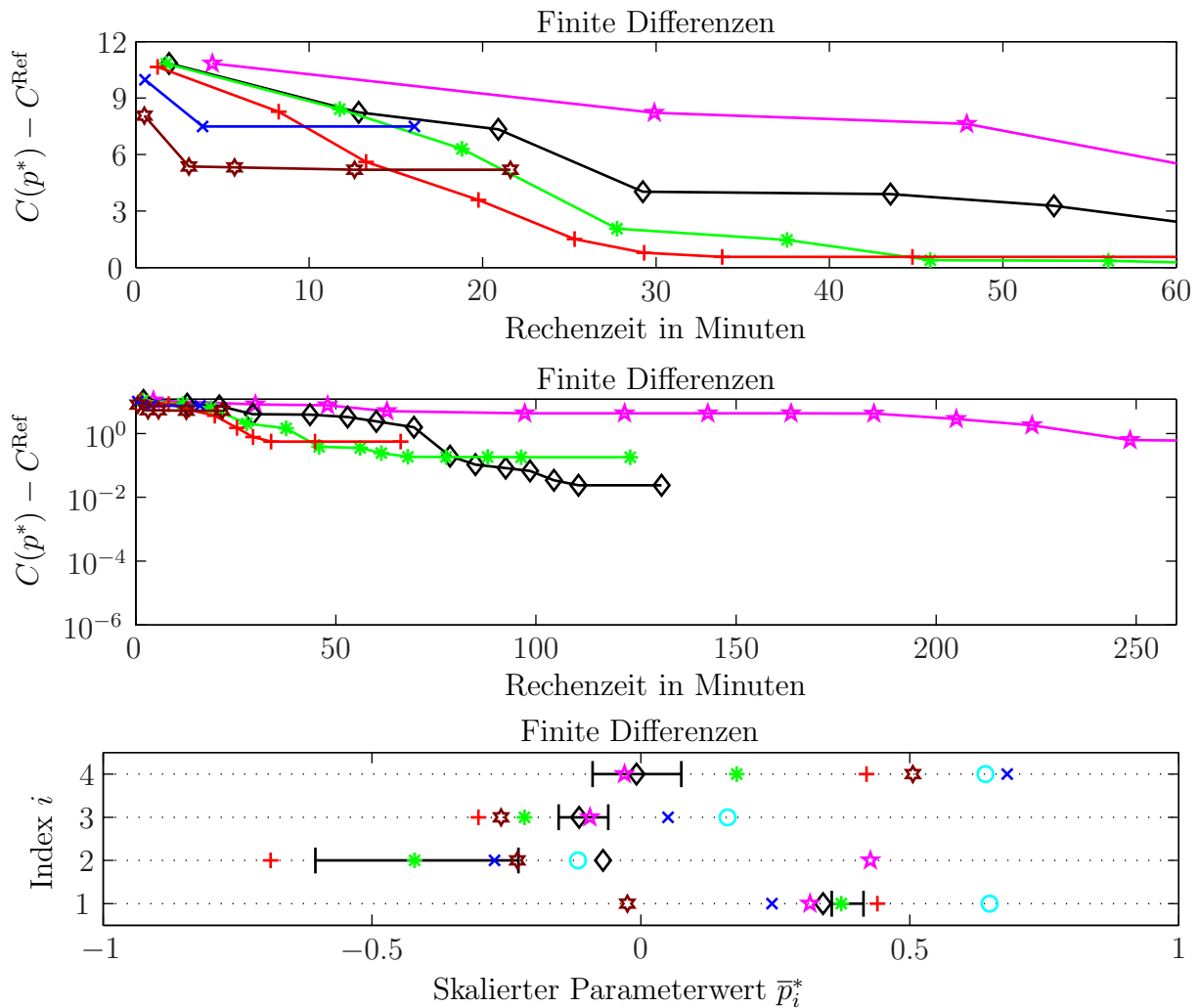


Abbildung C.6: Optimierung mit finiten Differenzen als Sensitivitäten. Oben, Mitte: Differenz zwischen Zielfunktion und Referenzwert. Unten: Identifizierte Parameterwerte. Farbgebung für  $\text{TOL}_{\text{Nom}}$  nach Abbildung 6.2.

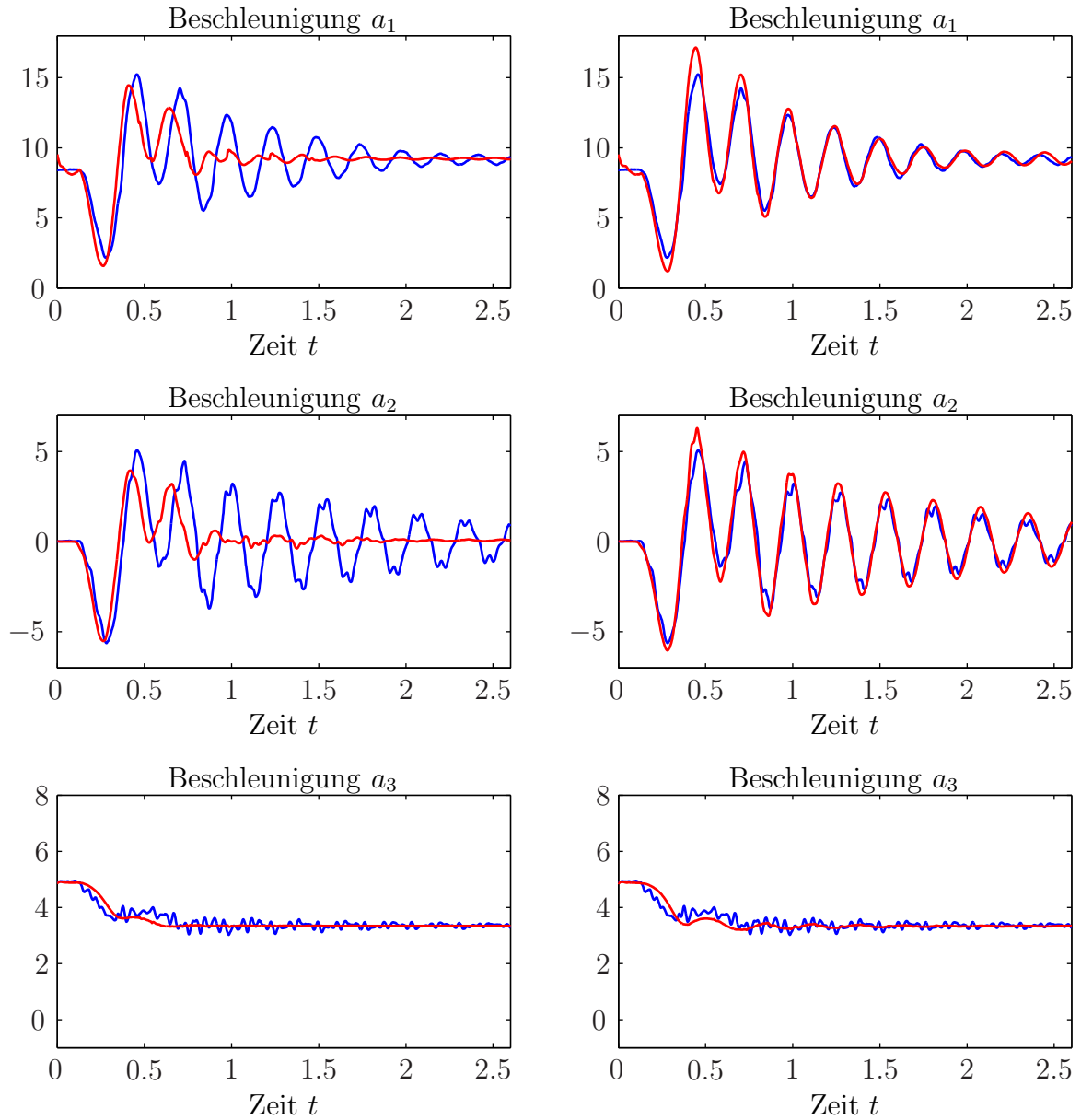


Abbildung C.7: Zeitverläufe der Beschleunigungen für  $p = p^0$  (links) und  $p = p^{\text{Ref}}$  (rechts). Andere Anregungsfunktionen als bei der Bestimmung von  $p^{\text{Ref}}$ . Fehler-toleranz  $\text{TOL} = 10^{-7}$ . **Blau**: Messungen. **Rot**: Simulationsergebnisse.

# Literaturverzeichnis

- [AC84] AUBIN, J. P. und A. CELLINA: *Differential Inclusions*. A Series of Comprehensive Studies in Mathematics. Springer-Verlag, 1984.
- [Åke07] ÅKESSON, J.: *Languages and Tools for Optimization of Large-Scale Systems*. PhD thesis, Department of Automatic Control, Lund University, Schweden, November 2007.
- [Arn98] ARNOLD, M.: *Zur Theorie und zur numerischen Lösung von Anfangswertproblemen für differentiell-algebraische Systeme von höherem Index*. Fortschrittsberichte VDI, Reihe 20, Nr. 264, VDI Verlag, 1998.
- [Arn01] ARNOLD, M.: *Reliable and efficient solution of sensitivity equations for differential algebraic systems*. Vortrag, ECMI-Workshop Numerical Methods in Multibody Dynamics, Bad Herrenalb, 2001.
- [AS98] ALEXANDER, J. C. und T. I. SEIDMAN: *Sliding Modes in Intersecting Switching Surfaces, I: Blending*. Houston Journal of Mathematics, 24(3):545–569, 1998.
- [BAF06] BACHMANN, B., P. ARONSSON und P. FRITZSON: *Robust Initialization of Differential Algebraic Equations*. In: *Proceedings of 5th International Modelica Conference*, Seiten 607–614, Wien, Österreich, September 2006.
- [Bar00] BARTON, P.: *Modeling, Simulation, and Sensitivity Analysis of Hybrid Systems*. In: *Proceedings of the IEEE International Symposium on Computer Aided Control System Design*, Seiten 117–122, September 2000.
- [Bau99] BAUER, I.: *Numerische Verfahren zur Lösung von Anfangswertaufgaben und zur Generierung von ersten und zweiten Ableitungen mit Anwendungen bei Optimierungsaufgaben in Chemie und Verfahrenstechnik*. Doktorarbeit, Naturwissenschaftlich-Mathematische Gesamtfakultät, Universität Heidelberg, 1999.
- [BBB<sup>+</sup>01] BINDER, T., L. BLANK, H. G. BOCK, R. BULIRSCH, W. DAHMEN, M. DIEHL, T. KRONSEDER, W. MARQUARDT, J. P. SCHLÖDER und O. VON STRYK: *Introduction to Model Based Optimization of Chemical Processes on Moving Horizons*. In: GRÖTSCHEL, M., S. O. KRUMKE und J. RAMBAU (Herausgeber): *Online Optimization of Large Scale Systems, State of the Art*, Seiten 295–340. Springer-Verlag, 2001.

- [BBV06] BISCHOF, C. H., H. M. BÜCKER und A. VEHRESCHILD: *A Macro Language for Derivative Definition in ADiMat*. In: BÜCKER, H. M., G. F. CORLISS, P. D. HOVLAND, U. NAUMANN und B. NORRIS (Herausgeber): *Automatic Differentiation: Applications, Theory, and Implementations*, Band 50 der Reihe *Lecture Notes in Computational Science and Engineering*, Seiten 181–188. Springer, Berlin, 2006.
- [BCKP94] BISCHOF, C., A. CARLE, P. KHADEMI und G. PUSCH: *Automatic Differentiation: Obtaining Fast and Reliable Derivatives – Fast*. In: *Proc. SIAM Symposium on Control Problems in Industry*, 1994.
- [BCP96] BRENNAN, K. E., S. L. CAMPBELL und L. R. PETZOLD: *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. Classics in Applied Mathematics, SIAM, Philadelphia, USA, 1996.
- [BHPS05] BALS, J., G. HOFER, A. PFEIFFER und C. SCHALLERT: *Virtual Iron Bird – A multidisciplinary modelling and simulation platform for new aircraft system architectures*. In: *Deutscher Luft- und Raumfahrtkongress 2005*, Friedrichshafen, September 2005.
- [Boc81] BOCK, H. G.: *Numerical treatment of inverse problems in chemical reaction kinetics*. In: EBERT, K. H., P. DEUFLHARD und W. JÄGER (Herausgeber): *Modelling of chemical reaction systems*, Band 18 der Reihe *Springer Series in Chemical Physics*, Seiten 102–125. 1981.
- [Boc87] BOCK, H. G.: *Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen*. Doktorarbeit, Bonner Mathematische Schriften Nr. 183, Bonn, 1987.
- [BP04] BRANDT-POLLMANN, U.: *Numerical solution of optimal control problems with implicitly defined discontinuities with applications in engineering*. Doktorarbeit, Naturwissenschaftlich-Mathematische Gesamtfakultät, Universität Heidelberg, 2004.
- [BRM96] BISCHOF, C., R. ROH und A. MAUER: *ADIC – An Extensible Automatic Differentiation Tool for ANSI-C*. Technical Report ANL/MCS-P626-1196, Mathematics and Computer Science Division, Argonne National Laboratory, USA, 1996.
- [Cal00] CALLIES, R.: *Entwurfsoptimierung und optimale Steuerung. Differential-algebraische Systeme, Mehrgitter-Mehrzielansätze und numerische Realisierung*. Habilitationsschrift, Zentrum Mathematik, Technische Universität München, Juli 2000.
- [Cel91] CELLIER, F. E.: *Continuous System Modeling*. Springer-Verlag, New York, 1991.
- [CEM<sup>+</sup>02] CLAUSS, C., H. ELMQVIST, S. E. MATTSSON, M. OTTER und P. SCHWARZ: *Mixed Domain Modeling in Modelica*. In: *Forum on Specification and Design Languages, FDL’02*, Marseille, Frankreich, September 2002.

- [CK06] CELLIER, F. E. und E. KOFMAN: *Continuous System Simulation*. Springer-Verlag, New York, 2006.
- [CLP02] CAO, Y., S. LI und L. PETZOLD: *Adjoint Sensitivity Analysis for Differential-Algebraic Equations: Algorithms and Software*. Journal of Computational and Applied Mathematics, 149(1):171–191, 2002.
- [CLPS03] CAO, Y., S. LI, L. PETZOLD und R. SERBAN: *Adjoint sensitivity analysis for differential-algebraic equations: the adjoint DAE system and its numerical solution*. SIAM Journal on Scientific Computing, 24(3):1076–1089, 2003.
- [DB02] DEUFLHARD, P. und F. BORNEMANN: *Numerische Mathematik II. Gewöhnliche Differentialgleichungen*. Walter de Gruyter, Berlin, New York, 2. Auflage, 2002.
- [Dym] DYMOLA: *Dynamic Modeling Laboratory*. Dynasim AB, Lund, Schweden, [www.dynasim.com](http://www.dynasim.com).
- [Eic92] EICH, E.: *Projizierende Mehrschrittverfahren zur numerischen Lösung von Bewegungsgleichungen technischer Mehrkörpersysteme mit Zwangsbedingungen und Unstetigkeiten*. Fortschrittberichte VDI, Reihe 18, Nr. 109, VDI Verlag, 1992.
- [EO94] ELMQVIST, H. und M. OTTER: *Methods for Tearing Systems of Equations in Object-Oriented Modeling*. In: *Proceedings ESM'94*, Seiten 326–332, Barcelona, Spanien, 1994.
- [ESF02] EICH-SOELLNER, E. und C. FÜHRER: *Numerical Methods in Multibody Dynamics*. B. G. Teubner, Stuttgart, Corrected reprint, 2002.
- [EUT06] ESCHENBACH, M., J. UNGETHÜM und P. TREFFINGER: *Vehicle model for transient simulation of a waste-heat-utilisation-unit containing extended PowerTrain and Fluid library components*. In: *Proceedings of 5th International Modelica Conference*, Seiten 405–410, Wien, Österreich, September 2006.
- [Fee98] FEEHERY, W. F.: *Dynamic Optimization with Path Constraints*. PhD thesis, Department of Chemical Engineering, Massachusetts Institute of Technology, USA, März 1998.
- [Füh88] FÜHRER, C.: *Differential-algebraische Gleichungssysteme in mechanischen Mehrkörpersystemen*. Doktorarbeit, Mathematisches Institut, Technische Universität München, 1988.
- [Fil88] FILIPPOV, A. F.: *Differential Equations with Discontinuous Righthand Sides*. Kluwer Academics Publishers, 1988.
- [FK00] FRANZ, S. und M. KIEHL: *Sensitivity Analysis of Initial Value Problems with Error Control*. Preprint 2110, Fachbereich Mathematik der TU Darmstadt, August 2000.
- [För07] FÖRG, M.: *Mehrkörpersysteme mit mengenwertigen Kraftgesetzen – Theorie und Numerik*. Fortschrittberichte VDI, Reihe 20, Nr. 411, VDI Verlag, 2007.

- [Fri04] FRITZSON, P.: *Principles of object-oriented modeling and simulation with Modelica 2.1*. IEEE Press, Wiley-Interscience, 2004.
- [Gea80] GEAR, C. W.: *Runge-Kutta-starters for multistep methods*. ACM Transactions on Mathematical Software, 6(3):263–279, 1980.
- [GFB99] GALÁN, S., W. F. FEEHERY und P. I. BARTON: *Parametric sensitivity functions for hybrid discrete/continuous systems*. Applied Numerical Mathematics, 31(1):17–47, 1999.
- [GGL85] GEAR, C. W., G. K. GUPTA und B. LEIMKUHLER: *Automatic integration of Euler-Lagrange equations with constraints*. Journal of Computational and Applied Mathematics, 12–13:77–90, 1985.
- [GJ91] GRÜBEL, G. und H.-D. JOOS: *RASP and RSYST – Two Complementary Program Libraries for Concurrent Control Engineering*. In: *Preprints of the 5th IFAC/IMACS Symposium CADCS'91*, Seiten 101–106, Swansea, U.K., 1991.
- [Glo01] GLOCKER, C.: *Set-Valued Force Laws. Dynamics of Non-Smooth Systems*. Springer-Verlag, 2001.
- [GM80] GETZ, W. M. und G. H. MARTIN: *Optimal Control Systems with State Variable Jump Discontinuities*. Journal of Optimization Theory and Applications, 31(2):195–205, 1980.
- [Gri00] GRIEWANK, A.: *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. Frontiers in Applied Mathematics, SIAM, Philadelphia, USA, 2000.
- [Gru96] GRUPP, F.: *Parameteridentifizierung nichtlinearer mechanischer Deskriptorsysteme mit Anwendungen in der Rad-Schiene-Dynamik*. Fortschrittberichte VDI, Reihe 8, Nr. 550, VDI Verlag, 1996.
- [HA97] HOLTZ, D. und J. S. ARORA: *An efficient implementation of adjoint sensitivity analysis for optimal control problems*. Structural Optimization, 13:223–229, 1997.
- [Hei92] HEIM, A.: *Parameteridentifizierung in differential-algebraischen Gleichungssystemen*. Diplomarbeit, Mathematisches Institut, Technische Universität München, November 1992.
- [Hin83] HINDMARSH, A. C.: *ODEPACK, A Systematized Collection of ODE Solvers*. IMACS Transaction on Scientific Computation, 1:55–64, 1983.
- [HM05] HAMANN, P. und V. MEHRMANN: *Numerical solution of hybrid differential-algebraic equations*. Preprint 263 MATHEON, DFG Research Center Mathematics for key technologies in Berlin, 2005.
- [HNW93] HAIRER, E., S. P. NØRSETT und G. WANNER: *Solving Ordinary Differential Equations I. Nonstiff problems*. Springer-Verlag, Second Revised Edition, 1993.

- [HP98] HISKENS, I. A. und M. A. PAI: *Sensitivity Analysis of Power System Trajectories: Recent Results*. In: *Proceedings of the IEEE International Symposium on Circuits and Systems*, Seiten 439–443 vol. 3, Monterey, Kalifornien, USA, 1998.
- [HS80] HIEBERT, K. L. und L. F. SHAMPINE: *Implicitly Defined Output Points for Solutions of ODE's*. Sandia Report SAND80-0180, USA, Februar 1980.
- [HW96] HAIRER, E. und G. WANNER: *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*. Springer-Verlag, Second Revised Edition, 1996.
- [Ise88a] ISERMANN, R.: *Identifikation dynamischer Systeme*, Band I: Frequenzgangmessung, Fourieranalyse, Korrelationsanalyse. Einführung in die Parameterschätzung. Springer-Verlag, Berlin, Heidelberg, New York, 1988.
- [Ise88b] ISERMANN, R.: *Identifikation dynamischer Systeme*, Band II: Parameterschätzmethoden, Kennwertermittlung und Modellabgleich, Zeitvariante, nichtlineare und Mehrgrößen-Systeme, Anwendungen. Springer-Verlag, Berlin, Heidelberg, New York, 1988.
- [JBL<sup>+</sup>02] JOOS, H.-D., J. BALS, G. LOOYE, K. SCHNEPPER und A. VARGA: *A multi-objective optimisation-based software environment for control system design*. In: *Proceedings of 2002 IEEE International Symposium on Computer Aided Control System Design*, Glasgow, Schottland, U. K., 2002.
- [Joo07] JOOS, H.-D.: *MOPS – Multi-Objective Parameter Synthesis, User's Guide V5.2*. DLR Deutsches Zentrum für Luft- und Raumfahrt, Institut für Robotik und Mechatronik Oberpfaffenhofen, 2007.
- [Kie98] KIEHL, M.: *Sensitivity Analysis of Stiff and Non-Stiff Initial-Value Problems*. International Series of Numerical Mathematics, Birkhäuser-Verlag, Basel, 124:143–152, 1998. Birkhäuser-Verlag, Basel.
- [Kli97] KLISCH, T.: *Kontaktmechanik in Starrkörpersystemen*. Shaker Verlag, Aachen, 1997.
- [KM06] KUNKEL, P. und V. MEHRMANN: *Differential-Algebraic Equations. Analysis and Numerical Solution*. EMS Publishing House, Zürich, 2006.
- [Kön97] KÖNIGSBERGER, K.: *Analysis 2*. Springer-Verlag, 2. Auflage, 1997.
- [KO06] KUHN, M. und M. OTTER: *Right Sized Multi Domain Simulation for Electrical Systems*. Vortrag, VIVACE Forum 2, Den Haag, Niederlande, Oktober 2006.
- [LH00] LEE, S. L. und P. D. HOVLAND: *Sensitivity Analysis Using Parallel ODE Solvers and Automatic Differentiation in C: SENSPVODE and ADIC*. In: *Proc. Automatic Differentiation: From Simulation to Optimization*, 2000.
- [LP99] LI, S. und L. R. PETZOLD: *Design of New DASPK for Sensitivity Analysis*. Technical Report, Department of Computer Science, University of California Santa Barbara, USA, Mai 1999.



- [LP02] LI, S. und L. R. PETZOLD: *Description of DASPKADJOINT: An Adjoint Sensitivity Solver for Differential-Algebraic Equations*. Technical Report, Department of Computer Science, University of California Santa Barbara, USA, 2002.
- [LTK<sup>+</sup>05] LOOYE, G., M. THÜMMEL, M. KURZE, M. OTTER und J. BALS: *Nonlinear Inverse Models for Control*. In: *Proceedings of the 4th International Modelica Conference*, Seiten 267–279, Hamburg, März 2005.
- [Maj98] MAJER, C. P.: *Parameterschätzung, Versuchsplanung und Trajektorienoptimierung für verfahrenstechnische Prozesse*. Fortschrittberichte VDI, Reihe 3, Nr. 538, VDI Verlag, 1998.
- [MKA00] MARTINS, J. R. R. A., I. M. KROO und J. J. ALONSO: *An Automated Method for Sensitivity Analysis Using Complex Variables*. In: *Proceedings of the 38th Aerospace Sciences Meeting*, Januar 2000.
- [Mod] MODELICA ASSOCIATION. [www.modelica.org](http://www.modelica.org).
- [Mod07] MODELICA ASSOCIATION: *Modelica - A Unified Object-Oriented Language for Physical Systems Modeling*. Language Specification, Version 3.0, September 2007.
- [MOE00] MATTSSON, S. E., H. OLSSON und H. ELMQVIST: *Dynamic Selection of States in Dymola*. In: *Proceedings Modelica Workshop*, Seiten 61–67, 2000.
- [Mor88] MOREAU, J. J.: *Unilateral Contact and Dry Friction in Finite Freedom Dynamics*. In: MOREAU, J. J. und PANAGIOTOPOULOS P. D. (Herausgeber): *Non-Smooth Mechanics and Applications*, CISM Courses and Lectures Vol. 302, Seiten 1–82. Springer-Verlag, 1988.
- [MP96] MALY, T. und L. PETZOLD: *Numerical Methods and Software for Sensitivity Analysis of Differential Algebraic Equations*. Applied Numerical Mathematics, 20(1):57–79, 1996.
- [MS93] MATTSSON, S. E. und G. SÖDERLIND: *Index reduction in differential-algebraic equations using dummy derivatives*. SIAM Journal of Scientific and Statistical Computing, 14:677–692, 1993.
- [MSA01] MARTINS, J. R. R. A., P. STURDZA und J. J. ALONSO: *The Connection Between the Complex-Step Derivative Approximation and Algorithmic Differentiation*. In: *Proceedings of the 39th Aerospace Sciences Meeting*, Januar 2001.
- [ND83] NOWAK, U. und P. DEUFLHARD: *Towards Parameter Identification for Large Chemical Reaction Systems*. In: DEUFLHARD, P. und E. HAIRER (Herausgeber): *Numerical Treatment of Inverse Problems in Differential and Integral Equations*, Seiten 13–26. Birkhäuser, Boston, 1983.
- [NW06] NOCEDAL, J. und S. J. WRIGHT: *Numerical Optimization*. Springer Series in Operations Research. Second Edition, 2006.

- [OEM99] OTTER, M., H. ELMQVIST und S. E. MATTSSON: *Hybrid Modeling in Modelica based on the Synchronous Data Flow Principle*. In: *Proceedings of the IEEE International Symposium on Computer Aided Control System Design*, Hawaii, USA, August 1999.
- [OEM07] OTTER, M., H. ELMQVIST und S. E. MATTSSON: *Multidomain Modeling with Modelica*. In: FISHWICK, P. (Herausgeber): *CRC Handbook of Dynamic System Modeling*, Kapitel 36, Seiten 36.1–36.27. CRC Press, 2007.
- [Ols98] OLSSON, H.: *Runge-Kutta Solution of Initial Value Problems: Methods, Algorithms and Implementation*. PhD thesis, Department of Computer Science, Lund University, Schweden, 1998.
- [OREM00] OTTER, M., M. REMELHE, S. ENGELL und P. MOSTERMAN: *Hybrid Models of Physical Systems and Discrete Controllers*. at – Automatisierungstechnik, 48:1–12, 2000.
- [Ott94] OTTER, M.: *Objektorientierte Modellierung mechatronischer Systeme am Beispiel geregelter Roboter*. Doktorarbeit, Fakultät für Maschinenbau, Ruhr-Universität Bochum, 1994.
- [Ott00] OTTER, M.: *Objektorientierte Modellierung Physikalischer Systeme*. at – Automatisierungstechnik, 47+48:A1–A68, 2000.
- [Ott01] OTTER, M.: *Objektorientierte Modellierung von Antriebssystemen*. In: SCHRÖDER, D. (Herausgeber): *Elektrische Antriebe – Regelung von Antriebssystemen*, Kapitel 20, Seiten 894–1009. Springer-Verlag, 2001.
- [Pan88] PANTELIDES, C.: *The Consistent Initialization of Differential-Algebraic Systems*. SIAM Journal of Scientific and Statistical Computing, Seiten 213–231, 1988.
- [Pel03] PELZ, G.: *Mechatronic Systems. Modelling and Simulation with HDLs*. Wiley, 2003.
- [Pet82a] PETZOLD, L. R.: *A description of DASSL: A differential/algebraic system solver*. Technical Report SAND82-8637, Sandia National Laboratories, Livermore, USA, 1982.
- [Pet82b] PETZOLD, L. R.: *Differential/Algebraic Equations are not ODE's*. SIAM Journal on Scientific Computing, 3(3):367–384, 1982.
- [PSO02] PELCHEN, C., C. SCHWEIGER und M. OTTER: *Modeling and Simulating the Efficiency of Gearboxes and of Planetary Gearboxes*. In: *Proceedings of 2nd International Modelica Conference*, Seiten 257–266, Oberpfaffenhofen, März 2002.
- [Rei07] REINER, M.: *Persönliche Mitteilung*, 2007.
- [Roz67] ROZENVASSER, E. N.: *General sensitivity equations of discontinuous systems*. Automat. Remote Control, Seiten 400–404, 1967.

- [SA05] STEWART, D. und M. ANITESCU: *Optimal Control of Systems with Discontinuous Differential Equations*. Preprint ANL/MCS-P1258-0605, Mathematics and Computer Science Division, Argonne National Laboratory, USA, Juni 2005.
- [SB95] SCHWERIN, R. VON und H. G. BOCK: *A Runge-Kutta starter for a multistep method for differential-algebraic systems with discontinuous effects*. Applied Numerical Mathematics, 18:337–350, 1995.
- [SBS98] SCHULZ, V. H., H. G. BOCK und M. C. STEINBACH: *Exploiting Invariants in the Numerical Solution of Multipoint Boundary Value Problems for DAEs*. SIAM Journal on Scientific Computing, 19(2):440–467, 1998.
- [Sch81] SCHITTKOWSKI, K.: *The Nonlinear Programming Method of Wilson, Han, and Powell with an Augmented Lagrangian Type Line Search Function, Part 1: Convergence Analysis*. Numerische Mathematik, 38:83–114, 1981.
- [Sch04a] SCHROEDER, K.: *Beiträge der Parameteridentifikation zur Fahrdynamikanalyse*. Fortschrittberichte VDI, Reihe 12, Nr. 564, VDI Verlag, 2004.
- [Sch04b] SCHUPP, G.: *Numerische Verzweigungsanalyse mit Anwendungen auf Rad-Schiene-Systeme*. Shaker Verlag, Aachen, 2004.
- [SCS00] SALTELLI, A., K. CHAN und E. M. SCOTT (Herausgeber): *Sensitivity Analysis*. Wiley Series in Probability and Statistics, 2000.
- [SH03] SERBAN, R. und A. C. HINDMARSH: *CVODES: An ODE Solver with Sensitivity Analysis Capabilities*. Preprint UCRL-JP-200039, Lawrence Livermore National Laboratory, USA, September 2003.
- [SPB06] SCHALLERT, C., A. PFEIFFER und J. BALS: *Generator Power Optimisation for a More-Electric Aircraft by use of a Virtual Iron Bird*. In: *ICAS Proceedings*, Hamburg, September 2006.
- [Spe93] SPELLUCCI, P.: *Numerische Verfahren der nichtlinearen Optimierung*. Birkhäuser Verlag, 1993.
- [ST96] STEWART, D. E. und J. C. TRINKLE: *An Implicit Time-Stepping Scheme for Rigid Body Dynamics with Inelastic Collision and Coulomb Friction*. International Journal for Numerical Methods in Engineering, 39:2637–2691, 1996.
- [ST00] SHAMPINE, L. F. und S. THOMPSON: *Event Location for Ordinary Differential Equations*. Computers and Mathematics with Applications, 39:43–54, 2000.
- [Ste00] STEWART, D. E.: *Rigid-Body Dynamics with Friction and Impact*. SIAM Review, 42(1):3–39, 2000.
- [SW95] STREHMEL, K. und R. WEINER: *Numerik gewöhnlicher Differentialgleichungen*. Teubner Studienbücher Mathematik. Teubner, Stuttgart, 1995.
- [Tar72] TARJAN, R. E.: *Depth First Search and Linear Graph Algorithms*. SIAM Journal of Computing, 1:146–160, 1972.

- 
- [Thü06] THÜMMEL, M.: *Modellbasierte Regelung mit nichtlinearen inversen Systemen und Beobachtern zur Optimierung der Dynamik von Robotern mit elastischen Gelenken*. Doktorarbeit, Technische Universität München, 2006.
- [Til01] TILLER, M.: *Introduction to Physical Modeling with Modelica*. Kluwer Academic Publishers, 2001.
- [Tis96] TISCHENDORF, C.: *Solution of index-2-DAEs and its application in circuit simulation*. Doktorarbeit, Humboldt-Universität zu Berlin, 1996.
- [TOB07] TOBOLAR, J., M. OTTER und T. BÜNTE: *Modelling of Vehicle Powertrains with the Modelica PowerTrain Library*. In: LASCHET, A. (Herausgeber): *Systemanalyse in der Kfz-Antriebstechnik IV*, Seiten 204–216. Expert-Verlag, 2007.
- [UGS99] UTKIN, V., J. GULDNER und J. SHI: *Sliding Mode Control in Electromechanical Systems*. CRC Press, 1999.
- [Wal03] WALLNER, S.: *Numerische Sensitivitätsanalyse in mechanischen Mehrkörpersystemen*. Diplomarbeit, Zentrum Mathematik, Technische Universität München, Oktober 2003.